

**P2S1521A**

**Programming Guidelines**

Copyright© 2005-2016. Socle Technology Corp. All Rights Reserved.

This document contains information that is confidential and proprietary to Socle Technology Corp. and may be disclosed only to those employees of Socle Technology with a need to know, or as otherwise permitted in writing by Socle Technology. Any copying, reproducing, modifying, use, or disclosure of this information (in whole or in part) which is not expressly permitted in writing by Socle Technology is strictly prohibited. At a minimum, this information is protected under trade secret, unfair competition, and copyright laws. Violations thereof may result in criminal penalties and fines.

Socle Technology reserves the right to change the information contained in this document to improve the function, design, or other aspects of this document. Socle Technology does not assume any liability for the application or use of this information, or for any error or omission in such information. Any warranties, whether express, statutory, implied, or otherwise, including but not limited to the implied warranties of merchantability or fitness for a particular purpose, are excluded. Any license under patent rights or any other intellectual property rights owned by Socle Technology or third parties shall be conveyed by Socle Technology in a separate license agreement between Socle Technology and the licensee.

**Trademark**

SoC\_ImP®, μPlatform,® and the Socle logo are the trademarks of Socle Technology. All other trademarks referred to herein are the property of their respective owners.

**DOCUMENT REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Date | By | Description |
| V0.6 | 2016/01/01 | Vincent Yen | Preliminary version |

TABLE OF CONTENTS

[**Trademark** 2](#_Toc450424512)

[**DOCUMENT REVISION HISTORY** 3](#_Toc450424513)

[TABLE OF CONTENTS 4](#_Toc450424514)

[1. SOC Programming Overview 26](#_Toc450424515)

[1.1 Programming Block Diagram 26](#_Toc450424516)

[1.2 Memory Map 26](#_Toc450424517)

[1.2.1 System Address Map 26](#_Toc450424518)

[1.2.2 Device Specific Memory Map 26](#_Toc450424519)

[1.2.3 Specific Function Register for IO Devices 27](#_Toc450424520)

[1.3 DMA Channel 28](#_Toc450424521)

[1.4 Interrupts Mapping 28](#_Toc450424522)

[2. Cortex-A7 31](#_Toc450424523)

[2.1 Register Summary 31](#_Toc450424524)

[2.1.1 Cortex-A7 c0 register summary 31](#_Toc450424525)

[2.1.2 Cortex-A7 c1 register summary 31](#_Toc450424526)

[2.1.3 Cortex-A7 c2 register summary 32](#_Toc450424527)

[2.1.4 Cortex-A7 c3 register summary 32](#_Toc450424528)

[2.1.5 Cortex-A7 c5 register summary 32](#_Toc450424529)

[2.1.6 Cortex-A7 c6 register summary 32](#_Toc450424530)

[2.1.7 Cortex-A7 c7 register summary 32](#_Toc450424531)

[2.1.8 Cortex-A7 c8 register summary 33](#_Toc450424532)

[2.1.9 Cortex-A7 c9 register summary 34](#_Toc450424533)

[2.1.10 Cortex-A7 c10 register summary 34](#_Toc450424534)

[2.1.11 Cortex-A7 c12 register summary 34](#_Toc450424535)

[2.1.12 Cortex-A7 c13 register summary 34](#_Toc450424536)

[2.1.13 Cortex-A7 c14 register summary 34](#_Toc450424537)

[2.1.14 Cortex-A7 c15 register summary 35](#_Toc450424538)

[2.1.15 64-bit CP15 system control register summary 35](#_Toc450424539)

[2.1.16 GIC memory map 35](#_Toc450424540)

[2.1.17 Distributor register summary 36](#_Toc450424541)

[2.1.18 CPU Interface register summary 37](#_Toc450424542)

[2.1.19 Virtual interface control register summary 37](#_Toc450424543)

[2.1.20 Virtual CPU interface register summary 37](#_Toc450424544)

[2.1.21 Address mapping for debug trace components 38](#_Toc450424545)

[2.1.22 CP14 debug Register summary 38](#_Toc450424546)

[2.1.23 PMU Register summary 40](#_Toc450424547)

[2.2 Register Descriptions 41](#_Toc450424548)

[3. System Control Unit 71](#_Toc450424549)

[3.1 Register Summary 71](#_Toc450424550)

[3.2 Detail Register Description 72](#_Toc450424551)

[4. Power Management Unit 92](#_Toc450424552)

[4.1 Register Summary 92](#_Toc450424553)

[4.2 Register Descriptions 92](#_Toc450424554)

[5. ISRAM Controller 102](#_Toc450424555)

[4.1 Register Summary 102](#_Toc450424560)

[4.2 Register Descriptions 103](#_Toc450424561)

[4.3 Programming Sequence 111](#_Toc450424562)

[6. CoreSight 114](#_Toc450424563)

[5.1 Register Summary 114](#_Toc450424565)

[5.1.1 APB Interconnect Register Summary 114](#_Toc450424566)

[5.1.2 APB-AP Register Summary 114](#_Toc450424567)

[5.1.3 Debug port register summary 115](#_Toc450424568)

[5.1.4 CTI Register Summary 115](#_Toc450424569)

[5.1.5 TPIU Register Summary 118](#_Toc450424570)

[5.1.6 TMC Register Summary 122](#_Toc450424571)

[5.1.7 TimeStamp Register Summary 123](#_Toc450424572)

[5.2 Register Descriptions 124](#_Toc450424573)

[5.2.1 APB Interconnect Register Descriptions 124](#_Toc450424574)

[5.2.2 APB-AP Register Descriptions 126](#_Toc450424575)

[5.2.3 Debug port Register Descriptions 128](#_Toc450424576)

[5.2.4 CTI Register Descriptions 130](#_Toc450424577)

[5.2.5 TPIU Register Descriptions 142](#_Toc450424578)

[5.2.6 TMC Register Descriptions 156](#_Toc450424579)

[5.2.7 TimeStamp Register Descriptions 167](#_Toc450424580)

[5.3 Programming Sequence 170](#_Toc450424581)

[7. SPI-NOR Controller(TBU) 172](#_Toc450424582)

[7.1 Register Summary 172](#_Toc450424583)

[7.2 Register Description 172](#_Toc450424584)

[7.3 Programming Sequence 179](#_Toc450424585)

[7.3.1 SPI master/slave controller operation flow chart 179](#_Toc450424586)

[7.3.2 Read Data 181](#_Toc450424587)

[8. IVA Acceleration 183](#_Toc450424588)

[8.1 Register Summary 183](#_Toc450424589)

[8.2 Register Descriptions 183](#_Toc450424590)

[8.2.1 SI Registers 183](#_Toc450424591)

[8.2.2 AVIF (IF\_CONFIG) 202](#_Toc450424592)

[8.2.3 GAMMA\_FE0\_MEM (GAMMA\_FE0\_MEM) 204](#_Toc450424593)

[8.2.4 HISTOGRAM\_MEM (HISTOGRAM\_MEM) 204](#_Toc450424594)

[8.2.5 CMD\_QUEUES (CMD\_QUEUES) 205](#_Toc450424595)

[8.2.6 AV\_CORE (IP\_CONFIG\_ANALYTIX) 205](#_Toc450424596)

[8.2.7 CONTROLLER 217](#_Toc450424597)

[8.2.8 CFM (IP\_CONFIG\_CFM) 218](#_Toc450424598)

[8.2.9 INTERRUPT\_CONTROLLER\_WRAPPER 224](#_Toc450424599)

[8.3 Programming Sequence 224](#_Toc450424600)

[9. Crypto Engine 226](#_Toc450424601)

[9.1 Register Summary 226](#_Toc450424602)

[9.1.1 SPDU Registers 229](#_Toc450424603)

[9.1.2 SPAcc core Registers 231](#_Toc450424604)

[9.1.3 PKA core Register 238](#_Toc450424605)

[9.1.4 TRNG Inside Register 242](#_Toc450424606)

[9.2 Programming Sequence 244](#_Toc450424607)

[10. DMA Engine 245](#_Toc450424608)

[10.1 Register Summary 245](#_Toc450424609)

[10.2 Register Descriptions 246](#_Toc450424610)

[10.3 Programming Sequence 258](#_Toc450424611)

[10.4 Programming Restrictions 260](#_Toc450424612)

[10.4.1 Fixed unaligned bursts 260](#_Toc450424613)

[10.4.2 Endian swap size restrictions 261](#_Toc450424614)

[10.4.3 Updating DMA channel control registers during a DMA cycle 261](#_Toc450424615)

[10.4.4 Instructions set summary 262](#_Toc450424616)

[10.4.5 Instructions 263](#_Toc450424617)

[11. GMAC Controller 275](#_Toc450424618)

[11.1 Register Summary 275](#_Toc450424619)

[11.2 Register Descriptions 275](#_Toc450424620)

[11.3 Programming Sequence 285](#_Toc450424621)

[12. SDIO Controller 287](#_Toc450424622)

[12.1 Register Summary 287](#_Toc450424623)

[12.2 Register Descriptions 288](#_Toc450424624)

[12.3 Programming Sequence 322](#_Toc450424625)

[12.3.1 Card Initialization 322](#_Toc450424626)

[12.3.2 Voltage Switching Sequence (UHS-I only) 324](#_Toc450424627)

[12.3.3 Clock Tuning procedure (UHS-I only) 325](#_Toc450424628)

[12.3.4 Commands Which Do Not Use DAT Line 325](#_Toc450424629)

[12.3.5 Commands with BUSY 326](#_Toc450424630)

[12.3.6 Commands Which Transfer Data on the DAT Line without DMA 328](#_Toc450424631)

[12.3.7 Commands Which Transfer Data on the DAT Line Using DMA 331](#_Toc450424632)

[13. DDR Controller + PHY 332](#_Toc450424633)

[13.1 Register Summary 332](#_Toc450424634)

[13.2 Register Descriptions 338](#_Toc450424635)

[13.2.1 Controller Register Descriptions 338](#_Toc450424636)

[13.2.2 PHY Register Descriptions 368](#_Toc450424637)

[13.3 Programming Sequence 373](#_Toc450424638)

[13.3.1 Controller Programming Sequence 373](#_Toc450424639)

[13.3.2 PHY Programming Sequence 375](#_Toc450424640)

[13.4 Programming Restrictions 377](#_Toc450424641)

[13.4.1 Controller Programming Restrictions 377](#_Toc450424642)

[14. I2C Controller 379](#_Toc450424643)

[14.1 Register Summary 379](#_Toc450424652)

[14.2 Register Descriptions 379](#_Toc450424653)

[14.3 Programming Sequence 382](#_Toc450424654)

[14.4 Programming Restrictions 389](#_Toc450424655)

[15. SPI Controller 390](#_Toc450424656)

[15.1 Register Summary 390](#_Toc450424657)

[15.2 Register Description 390](#_Toc450424658)

[15.3 Programming Sequence 397](#_Toc450424659)

[15.3.1 HW command timing 397](#_Toc450424660)

[15.3.2 SPI master controller operation flow chart 398](#_Toc450424661)

[15.3.3 SPI slave controller operation flow chart 399](#_Toc450424662)

[15.4 Programming Restrictions 400](#_Toc450424663)

[16. UART Controller 401](#_Toc450424664)

[16.1 Registers Summary 401](#_Toc450424665)

[16.2 Registers Descriptions 401](#_Toc450424666)

[16.3 Programming Sequence 406](#_Toc450424667)

[16.4 Programming Restrictions 408](#_Toc450424668)

[17. Watch Timer 409](#_Toc450424669)

[17.1 Register Summary 409](#_Toc450424670)

[17.2 Register Descriptions 409](#_Toc450424671)

[17.3 Programming Sequence 409](#_Toc450424672)

[17.4 Programming Restrictions 410](#_Toc450424673)

[18. Timer 411](#_Toc450424674)

[18.1 Register Summary 411](#_Toc450424675)

[18.2 Register Descriptions 411](#_Toc450424676)

[18.3 Programming Sequence 412](#_Toc450424677)

[18.4 Programming Restrictions 412](#_Toc450424678)

[19. GPIO Controller 413](#_Toc450424679)

[19.1 Register Summary 413](#_Toc450424680)

[19.2 Register Descriptions 413](#_Toc450424681)

[19.3 Programming Sequence 415](#_Toc450424682)

[19.3.1 Rising Edge Detected 415](#_Toc450424683)

[19.3.2 Falling Edge Detected 416](#_Toc450424684)

[19.3.3 Both Edges Detected 416](#_Toc450424685)

[19.3.4 High Level Detected 417](#_Toc450424686)

[19.3.5 Low Level Detected 417](#_Toc450424687)

[19.4 Programming Restrictions 418](#_Toc450424688)

[20. PWM 419](#_Toc450424689)

[20.1 Register Summary 419](#_Toc450424690)

[20.2 Register Descriptions 419](#_Toc450424691)

[20.3 Programming Sequence 420](#_Toc450424692)

[20.4 Programming Restrictions 420](#_Toc450424693)

[21. IOMUX 421](#_Toc450424694)

[21.1 Register Summary 421](#_Toc450424695)

[21.2 Register Descriptions 423](#_Toc450424696)

[21.2.1 Pad Control Register 423](#_Toc450424697)

[21.2.2 SW MUX Select Register 472](#_Toc450424698)

[21.3 IOMUX Table 474](#_Toc450424699)

[21.3.1 Normal & Engineering & FPGA Mode 474](#_Toc450424700)

[21.3.2 DFT & MBIST Mode 476](#_Toc450424701)

[21.3.3 Analog Test Mode 479](#_Toc450424702)

[21.3.4 NAND-Tree Test Mode 482](#_Toc450424703)

[21.3.5 Debug Mode 484](#_Toc450424704)

[21.4 Programming Configuration 489](#_Toc450424705)

[21.5 Output Timing 489](#_Toc450424706)

List of Tables

[Table 1.System memory map 26](#_Toc450425584)

[Table 2.Device memory map 27](#_Toc450425585)

[Table 3.Register for IO devices 27](#_Toc450425586)

[Table 4.DMA peripheral 28](#_Toc450425587)

[Table 5.Interrupts Mapping 28](#_Toc450425588)

[Table 6.Cortex-A7 c0 register summary 31](#_Toc450425589)

[Table 7.Cortex-A7 c1 register summary 31](#_Toc450425590)

[Table 8.Cortex-A7 c2 register summary 32](#_Toc450425591)

[Table 9.Cortex-A7 c3 register summary 32](#_Toc450425592)

[Table 10.Cortex-A7 c5 register summary 32](#_Toc450425593)

[Table 11.Cortex-A7 c6 register summary 32](#_Toc450425594)

[Table 12.Cortex-A7 c7 register summary 32](#_Toc450425595)

[Table 13.Cortex-A7 c8 register summary 33](#_Toc450425596)

[Table 14.Cortex-A7 c9 register summary 34](#_Toc450425597)

[Table 15.Cortex-A7 c10 register summary 34](#_Toc450425598)

[Table 16.Cortex-A7 c12 register summary 34](#_Toc450425599)

[Table 17.Cortex-A7 c13 register summary 34](#_Toc450425600)

[Table 18.Cortex-A7 c14 register summary 35](#_Toc450425601)

[Table 19.Cortex-A7 c15 register summary 35](#_Toc450425602)

[Table 20. 64-bit CP15 system control register summary 35](#_Toc450425603)

[Table 21. CP15 GIC memory map 35](#_Toc450425604)

[Table 22. Distributor register summary 36](#_Toc450425605)

[Table 23. CPU Interface register summary 37](#_Toc450425606)

[Table 24. Virtual interface control register summary 37](#_Toc450425607)

[Table 25. Virtual CPU interface register summary 37](#_Toc450425608)

[Table 26. Address mapping for debug trace components 38](#_Toc450425609)

[Table 27. CP14 debug Register summary 38](#_Toc450425610)

[Table 28. PMU Register summary 40](#_Toc450425611)

[Table 29. MIDR bit assignments 41](#_Toc450425612)

[Table 30. CTR bit assignments 41](#_Toc450425613)

[Table 31. TLBTR bit assignments 41](#_Toc450425614)

[Table 32. MPIDR bit assignments 42](#_Toc450425615)

[Table 33. REVIDR bit assignments 42](#_Toc450425616)

[Table 34.ID\_PFR0 bit assignments 42](#_Toc450425617)

[Table 35. ID\_PFR1 bit assignments 42](#_Toc450425618)

[Table 36. ID\_DFR0 bit assignments 43](#_Toc450425619)

[Table 37. ID\_MMFR0 bit assignments 43](#_Toc450425620)

[Table 38. ID\_MMFR1 bit assignments 43](#_Toc450425621)

[Table 39. ID\_MMFR2 bit assignments 44](#_Toc450425622)

[Table 40. ID\_MMFR3 bit assignments 44](#_Toc450425623)

[Table 41. ID\_ISAR0 bit assignments 45](#_Toc450425624)

[Table 42. ID\_PFR1 bit assignments 45](#_Toc450425625)

[Table 43. ID\_DFR0 bit assignments 46](#_Toc450425626)

[Table 44. ID\_MMFR0 bit assignments 46](#_Toc450425627)

[Table 45. ID\_MMFR1 bit assignments 46](#_Toc450425628)

[Table 46. ID\_MMFR2 bit assignments 47](#_Toc450425629)

[Table 47. ID\_MMFR3 bit assignments 47](#_Toc450425630)

[Table 48. ID\_ISAR0 bit assignments 48](#_Toc450425631)

[Table 49. PRRR bit assignments 48](#_Toc450425632)

[Table 50. MAIR0 and MAIR1 bit assignments 49](#_Toc450425633)

[Table 51. NMRR bit assignments 49](#_Toc450425634)

[Table 52. ACTLR bit assignments 49](#_Toc450425635)

[Table 53. CPACR bit assignments 50](#_Toc450425636)

[Table 54. SCR bit assignments 51](#_Toc450425637)

[Table 55. NSACR bit assignments 52](#_Toc450425638)

[Table 56. HSCTLR bit assignments 53](#_Toc450425639)

[Table 57. HDCR bit assignments 53](#_Toc450425640)

[Table 58. HCPTR bit assignments 54](#_Toc450425641)

[Table 59. DFSR bit assignments for Short-descriptor translation table format 55](#_Toc450425642)

[Table 60. DFSR bit assignments for Long-descriptor translation table format 56](#_Toc450425643)

[Table 61. IFSR bit assignments for short-descriptor translation table format 57](#_Toc450425644)

[Table 62. IFSR bit assignments for Long-descriptor translation table format 57](#_Toc450425645)

[Table 63. HSR bit assignments 57](#_Toc450425646)

[Table 64. L2CTLR bit assignments 58](#_Toc450425647)

[Table 65. L2ECTLR bit assignments 58](#_Toc450425648)

[Table 66. CBAR bit assignments 58](#_Toc450425649)

[Table 67. GICD\_TYPER bit assignments 58](#_Toc450425650)

[Table 68. GICD\_IIDR bit assignments 59](#_Toc450425651)

[Table 69. GICD\_PPISR bit assignments 59](#_Toc450425652)

[Table 70. GICD\_SPISRn bit assignments 59](#_Toc450425653)

[Table 71. GICC\_IIDR bit assignments 60](#_Toc450425654)

[Table 72. GICH\_VTR bit assignments 60](#_Toc450425655)

[Table 73. DBGDIDR bit assignments 60](#_Toc450425656)

[Table 74. DBGPCSR bit assignments 61](#_Toc450425657)

[Table 75. DBGDRCR bit assignments 61](#_Toc450425658)

[Table 76. DBGEACR bit assignments 61](#_Toc450425659)

[Table 77. DBGBVR bit assignments when register is used for address comparison 62](#_Toc450425660)

[Table 78. DBGBVR bit assignments when register is used for Context ID comparison 62](#_Toc450425661)

[Table 79. DBGBCR bit assignments 62](#_Toc450425662)

[Table 80. DBGWVR bit assignments 63](#_Toc450425663)

[Table 81. DBGWCR bit assignments 63](#_Toc450425664)

[Table 82. DBGDRAR bit assignments 64](#_Toc450425665)

[Table 83. DBGBXVR bit assignments 65](#_Toc450425666)

[Table 84. DBGOSLAR bit assignments 65](#_Toc450425667)

[Table 85. DBGOSLSR bit assignments 65](#_Toc450425668)

[Table 86. DBGPRCR bit assignments 65](#_Toc450425669)

[Table 87. DBGDSAR bit assignments 66](#_Toc450425670)

[Table 88. DBGITMISCOUT bit assignments 66](#_Toc450425671)

[Table 89. DBGITMISCIN bit assignments 67](#_Toc450425672)

[Table 90. DBGITCTRL bit assignments 67](#_Toc450425673)

[Table 91. DBGCLAIMSET bit assignments 67](#_Toc450425674)

[Table 92. DBGCLAIMCLR bit assignments 67](#_Toc450425675)

[Table 93. DBGDEVID1 bit assignments 67](#_Toc450425676)

[Table 94. DBGDEVID bit assignments 68](#_Toc450425677)

[Table 95. Debug Peripheral ID 4/5/6/7/0/1/2/3 68](#_Toc450425678)

[Table 96. Summary of the Component Identification Registers 68](#_Toc450425679)

[Table 97. PMCR bit assignments 69](#_Toc450425680)

[Table 98. The performance monitors peripheral identification registers 4/5/6/7/0/1/2/3 69](#_Toc450425681)

[Table 99. Summary of the Component Identification Registers 70](#_Toc450425682)

[Table 100.SoC register summary 71](#_Toc450425683)

[Table 101.Chip ID Register 72](#_Toc450425684)

[Table 102.System memory mapping register 72](#_Toc450425685)

[Table 103.RCU control register 72](#_Toc450425686)

[Table 104.RCU software reset register 72](#_Toc450425687)

[Table 105.RCU peripheral software reset 73](#_Toc450425688)

[Table 106.RCU reset status register 73](#_Toc450425689)

[Table 107.RCU General Purpose Register0 74](#_Toc450425690)

[Table 108.CMU Control Register 74](#_Toc450425691)

[Table 109.CMU Status Register 74](#_Toc450425692)

[Table 110.CMU CPU PLL Configuration Register 0 75](#_Toc450425693)

[Table 111.CMU CPU PLL Configuration Register 1 75](#_Toc450425694)

[Table 112.CMU CPU PLL Configuration Register2 75](#_Toc450425695)

[Table 113.CMU CPU PLL Configuration Register 0 76](#_Toc450425696)

[Table 114.CMU DDR PLL Configuration Register1 76](#_Toc450425697)

[Table 115.CMU DDR PLL Configuration Register2 77](#_Toc450425698)

[Table 116.CMU system PLL Configuration Register0 77](#_Toc450425699)

[Table 117.CMU system PLL Configuration Register1 78](#_Toc450425700)

[Table 118.CMU SYS PLL Configuration Register2 78](#_Toc450425701)

[Table 119.CMU VIDEO PLL Configuration Register 0 78](#_Toc450425702)

[Table 120.CMU VIDEO PLL Configuration Register 1 79](#_Toc450425703)

[Table 121.CMU VIDEO PLL Configuration Register 2 79](#_Toc450425704)

[Table 122.CMU Clock Source Select Register 79](#_Toc450425705)

[Table 123.CMU MUX Select Setting Register 80](#_Toc450425706)

[Table 124.CMU SYS Domain Divider Setting Register 80](#_Toc450425707)

[Table 125.CMU Debug Domain Divider Setting Register 81](#_Toc450425708)

[Table 126.CMU Other Domain Divider Setting Register 82](#_Toc450425709)

[Table 127.CMU MUX Handshake In-Process Register 82](#_Toc450425710)

[Table 128.CMU Divider Handshake In-Process Register 83](#_Toc450425711)

[Table 129.CMU SYS Domain Clock Gating Register 85](#_Toc450425712)

[Table 130.CMU DBG Domain Clock Gating Register 86](#_Toc450425713)

[Table 131.CMU PIO Domain Clock Gating Register 86](#_Toc450425714)

[Table 132.CMU Other Domain Clock Gating Register 86](#_Toc450425715)

[Table 133.CMU NOC Clock Gating Register 86](#_Toc450425716)

[Table 134.ISRAM Control Register0 87](#_Toc450425717)

[Table 135.ISRAM Control Register1 88](#_Toc450425718)

[Table 136.DMAC Control Register0 88](#_Toc450425719)

[Table 137.DMAC Control Register1 88](#_Toc450425720)

[Table 138.DDR Control Register 88](#_Toc450425721)

[Table 139.DDR Control/Status Register 88](#_Toc450425722)

[Table 140.Crypto Control Register 89](#_Toc450425723)

[Table 141.GMAC Control Register 89](#_Toc450425724)

[Table 142.SPINOR Control Register 89](#_Toc450425725)

[Table 143.SDIO Control/status Register 89](#_Toc450425726)

[Table 144.CA7 Options register0 90](#_Toc450425727)

[Table 145.CA7 Options register1 90](#_Toc450425728)

[Table 146.CA7 Options register2 90](#_Toc450425729)

[Table 147.CA7 Options register3 90](#_Toc450425730)

[Table 148.CoreSight Control Register 91](#_Toc450425731)

[Table 149.CoreSight Control Register 91](#_Toc450425732)

[Table 150.PMU Register summary 92](#_Toc450425733)

[Table 151.ISRAM Register summary 102](#_Toc450425734)

[Table 152.APB Interconnect Register summary 114](#_Toc450425735)

[Table 153.APB-AP Register summary 114](#_Toc450425736)

[Table 154. CTI Register summary 115](#_Toc450425737)

[Table 155. TPIU Register summary 118](#_Toc450425738)

[Table 156. TMC Register summary 122](#_Toc450425739)

[Table 157. TPIU Register summary 123](#_Toc450425740)

[Table 158. ROM table entry register 124](#_Toc450425741)

[Table 159. Peripheral ID4 Register 124](#_Toc450425742)

[Table 160. Peripheral ID5-7 Registers 125](#_Toc450425743)

[Table 161. Peripheral ID0 Register 125](#_Toc450425744)

[Table 162. Peripheral ID1 Register 125](#_Toc450425745)

[Table 163. Peripheral ID2 Register 125](#_Toc450425746)

[Table 164. Peripheral ID3 Register 125](#_Toc450425747)

[Table 165. Component ID0 Register 126](#_Toc450425748)

[Table 166. Component ID1 Register 126](#_Toc450425749)

[Table 167. Component ID2 Register 126](#_Toc450425750)

[Table 168. Component ID3 Register 126](#_Toc450425751)

[Table 169. APB-AP Control/Status Word Register 126](#_Toc450425752)

[Table 170. APB-AP Transfer Address Register 127](#_Toc450425753)

[Table 171. APB-AP Data Read/Write Register 127](#_Toc450425754)

[Table 172. APB-AP Banked Data Registers, BD0-BD3 127](#_Toc450425755)

[Table 173. Debug APB ROM Address 128](#_Toc450425756)

[Table 174. APB-AP Identification Register 128](#_Toc450425757)

[Table 175. AP Abort Register 128](#_Toc450425758)

[Table 176. Identification Code Register 128](#_Toc450425759)

[Table 177. Control/Status Register, CTRL/STAT 128](#_Toc450425760)

[Table 178. AP Select Register 129](#_Toc450425761)

[Table 179. Data Link Control Register 130](#_Toc450425762)

[Table 180. Data Link Protocol Identification Register 130](#_Toc450425763)

[Table 181. CTI Control Register 130](#_Toc450425764)

[Table 182. CTI Interrupt Acknowledge Register 130](#_Toc450425765)

[Table 183. CTI Application Trigger Set Register 131](#_Toc450425766)

[Table 184. CTI Application Trigger Clear Register 131](#_Toc450425767)

[Table 185. CTI Application Pulse Register 131](#_Toc450425768)

[Table 186. CTI Trigger 0 to Channel Enable Register 131](#_Toc450425769)

[Table 187. CTI Trigger 1 to Channel Enable Register 131](#_Toc450425770)

[Table 188. CTI Trigger 2 to Channel Enable Register 132](#_Toc450425771)

[Table 189. CTI Trigger 3 to Channel Enable Register 132](#_Toc450425772)

[Table 190. CTI Trigger 4 to Channel Enable Register 132](#_Toc450425773)

[Table 191. CTI Trigger 5 to Channel Enable Register 132](#_Toc450425774)

[Table 192. CTI Trigger 6 to Channel Enable Register 133](#_Toc450425775)

[Table 193. CTI Trigger 7 to Channel Enable Register 133](#_Toc450425776)

[Table 194. CTI Channel to Trigger 0 Enable Register 133](#_Toc450425777)

[Table 195. CTI Channel to Trigger 1 Enable Register 134](#_Toc450425778)

[Table 196. CTI Channel to Trigger 2 Enable Register 134](#_Toc450425779)

[Table 197. CTI Channel to Trigger 3 Enable Register 134](#_Toc450425780)

[Table 198. CTI Channel to Trigger 4 Enable Register 134](#_Toc450425781)

[Table 199. CTI Channel to Trigger 5 Enable Register 135](#_Toc450425782)

[Table 200. CTI Channel to Trigger 6 Enable Register 135](#_Toc450425783)

[Table 201. CTI Channel to Trigger 7 Enable Register 135](#_Toc450425784)

[Table 202. CTI Trigger In Status Register 135](#_Toc450425785)

[Table 203. CTI Trigger Out Status Register 136](#_Toc450425786)

[Table 204. CTI Channel In Status Register 136](#_Toc450425787)

[Table 205. CTI Channel Out Status Register 136](#_Toc450425788)

[Table 206. Enable CTI Channel Gate Register 136](#_Toc450425789)

[Table 207. External Multiplexer Control Register 136](#_Toc450425790)

[Table 208. ITCHINACK Register 137](#_Toc450425791)

[Table 209. ITCHOUT Register 137](#_Toc450425792)

[Table 210. ITTRIGOUT Register 137](#_Toc450425793)

[Table 211. ITCHOUTACK Register 137](#_Toc450425794)

[Table 212. ITCHIN Register 137](#_Toc450425795)

[Table 213. ITTRIGIN Register 137](#_Toc450425796)

[Table 214. Integration Mode Control Register 137](#_Toc450425797)

[Table 215. Claim Tag Set Register 138](#_Toc450425798)

[Table 216. Claim Tag Clear Register 138](#_Toc450425799)

[Table 217. Lock Access Register 138](#_Toc450425800)

[Table 218. Lock Status Register 138](#_Toc450425801)

[Table 219. Authentication Status Register 139](#_Toc450425802)

[Table 220. Device Configuration Register 139](#_Toc450425803)

[Table 221. Device Type Identifier Register 139](#_Toc450425804)

[Table 222. Peripheral ID4 Register 139](#_Toc450425805)

[Table 223. Peripheral ID5-7 Registers 140](#_Toc450425806)

[Table 224. Peripheral ID0 Register 140](#_Toc450425807)

[Table 225. Peripheral ID1 Register 140](#_Toc450425808)

[Table 226. Peripheral ID2 Register 140](#_Toc450425809)

[Table 227. Peripheral ID3 Register 141](#_Toc450425810)

[Table 228. Component ID0 Register 141](#_Toc450425811)

[Table 229. Component ID1 Register 141](#_Toc450425812)

[Table 230. Component ID2 Register 141](#_Toc450425813)

[Table 231. Component ID3 Register 141](#_Toc450425814)

[Table 232. Supported Port Size Register 142](#_Toc450425815)

[Table 233. Current Port Size Register 144](#_Toc450425816)

[Table 234. Supported Trigger Modes Register 146](#_Toc450425817)

[Table 235. Trigger Counter Value Register 147](#_Toc450425818)

[Table 236. Trigger Multiplier Register 147](#_Toc450425819)

[Table 237. Supported Test Patterns/Modes Register 147](#_Toc450425820)

[Table 238. Current Test Pattern/Modes Register 148](#_Toc450425821)

[Table 239. TPIU Test Pattern Repeat Counter Register 148](#_Toc450425822)

[Table 240. Formatter and Flush Status Register 148](#_Toc450425823)

[Table 241. Formatter and Flush Control Register 149](#_Toc450425824)

[Table 242. Formatter Synchronization Counter Register 150](#_Toc450425825)

[Table 243. TPIU EXCTL Port Register IN 150](#_Toc450425826)

[Table 244. TPIU EXCTL Port Register Out 150](#_Toc450425827)

[Table 245. Integration Test Trigger In and Flush In Acknowledge Register 150](#_Toc450425828)

[Table 246. Integration Test Trigger In and Flush In Register 150](#_Toc450425829)

[Table 247. Integration Test ATB Data Register 0 150](#_Toc450425830)

[Table 248. Integration Test ATB Control Register 2 151](#_Toc450425831)

[Table 249. Integration Test ATB Control Register 1 151](#_Toc450425832)

[Table 250. Integration Test ATB Control Register 0 151](#_Toc450425833)

[Table 251. Integration Mode Control Register 151](#_Toc450425834)

[Table 252. Claim Tag Set Register 152](#_Toc450425835)

[Table 253. Claim Tag Clear Register 152](#_Toc450425836)

[Table 254. Lock Access Register 152](#_Toc450425837)

[Table 255. Lock Status Register 152](#_Toc450425838)

[Table 256. Authentication Status Register 153](#_Toc450425839)

[Table 257. Device Configuration Register 153](#_Toc450425840)

[Table 258. Device Type Identifier Register 153](#_Toc450425841)

[Table 259. Peripheral ID4 Register 153](#_Toc450425842)

[Table 260. Peripheral ID5-7 Registers 154](#_Toc450425843)

[Table 261. Peripheral ID0 Register 154](#_Toc450425844)

[Table 262. Peripheral ID1 Register 154](#_Toc450425845)

[Table 263. Peripheral ID2 Register 154](#_Toc450425846)

[Table 264. Peripheral ID3 Register 155](#_Toc450425847)

[Table 265. Component ID0 Register 155](#_Toc450425848)

[Table 266. Component ID1 Register 155](#_Toc450425849)

[Table 267. Component ID2 Register 155](#_Toc450425850)

[Table 268. Component ID3 Register 156](#_Toc450425851)

[Table 269. TMC RAM Depth Register 156](#_Toc450425852)

[Table 270. TMC Status Register 156](#_Toc450425853)

[Table 271. TMC RAM Read Data Register 156](#_Toc450425854)

[Table 272. TMC RAM Read Pointer Register 156](#_Toc450425855)

[Table 273. TMC RAM Write Pointer Register 157](#_Toc450425856)

[Table 274. TMC Trigger Counter Register 157](#_Toc450425857)

[Table 275. TMC Control Register 157](#_Toc450425858)

[Table 276. TMC RAM Write Data Register 157](#_Toc450425859)

[Table 277. Latched Buffer Fill Level Register 158](#_Toc450425860)

[Table 278. Current Buffer Fill Level Register 158](#_Toc450425861)

[Table 279. BUFWM Register 158](#_Toc450425862)

[Table 280. RRPHI Register 158](#_Toc450425863)

[Table 281. RWPHI Register 158](#_Toc450425864)

[Table 282. AXICTL Register 158](#_Toc450425865)

[Table 283. DBALO Register 159](#_Toc450425866)

[Table 284. DBAHI Register 160](#_Toc450425867)

[Table 285. TMC Formatter and Flush Status Register 160](#_Toc450425868)

[Table 286. TMC Formatter and Flush Control Register 160](#_Toc450425869)

[Table 287. Integration Test Miscellaneous Output Register 0 161](#_Toc450425870)

[Table 288. Integration Test Trigger In and Flush In Acknowledge Register 161](#_Toc450425871)

[Table 289. Integration Test Trigger In and Flush In Register 161](#_Toc450425872)

[Table 290. Integration Test ATB Data Register 0 162](#_Toc450425873)

[Table 291. Integration Test ATB Control Register 2 162](#_Toc450425874)

[Table 292. Integration Test ATB Control Register 1 162](#_Toc450425875)

[Table 293. Integration Test ATB Control Register 0 162](#_Toc450425876)

[Table 294. Integration Mode Control Register 162](#_Toc450425877)

[Table 295. Claim Tag Set Register 163](#_Toc450425878)

[Table 296. Claim Tag Clear Register 163](#_Toc450425879)

[Table 297. Lock Access Register 163](#_Toc450425880)

[Table 298. Lock Status Register 163](#_Toc450425881)

[Table 299. Authentication Status Register 164](#_Toc450425882)

[Table 300. Device Configuration Register 164](#_Toc450425883)

[Table 301. Device Type Identifier Register 164](#_Toc450425884)

[Table 302. Peripheral ID4 Register 164](#_Toc450425885)

[Table 303. Peripheral ID5-7 Registers 165](#_Toc450425886)

[Table 304. Peripheral ID0 Register 165](#_Toc450425887)

[Table 305. Peripheral ID1 Register 165](#_Toc450425888)

[Table 306. Peripheral ID2 Register 165](#_Toc450425889)

[Table 307. Peripheral ID3 Register 166](#_Toc450425890)

[Table 308. Component ID0 Register 166](#_Toc450425891)

[Table 309. Component ID1 Register 166](#_Toc450425892)

[Table 310. Component ID2 Register 166](#_Toc450425893)

[Table 311. Component ID3 Register 166](#_Toc450425894)

[Table 312. Counter Control Register 167](#_Toc450425895)

[Table 313. Counter Status Register 167](#_Toc450425896)

[Table 314. Current Counter Value Lower register 167](#_Toc450425897)

[Table 315. CNTFID0 Register 167](#_Toc450425898)

[Table 316. CNTCVU Register 167](#_Toc450425899)

[Table 317. Peripheral ID4 Register 167](#_Toc450425900)

[Table 318. Peripheral ID5-7 Registers 168](#_Toc450425901)

[Table 319. Peripheral ID0 Register 168](#_Toc450425902)

[Table 320. Peripheral ID1 Register 168](#_Toc450425903)

[Table 321. Peripheral ID2 Register 169](#_Toc450425904)

[Table 322. Peripheral ID3 Register 169](#_Toc450425905)

[Table 323. Component ID0 Register 169](#_Toc450425906)

[Table 324. Component ID1 Register 169](#_Toc450425907)

[Table 325. Component ID2 Register 169](#_Toc450425908)

[Table 326. Component ID3 Register 170](#_Toc450425909)

[Table 327.SPINOR Controller Register summary 172](#_Toc450425910)

[Table 328.SPI\_TxR 172](#_Toc450425911)

[Table 329.SPI\_RxR 172](#_Toc450425912)

[Table 330.SPI\_IER 172](#_Toc450425913)

[Table 331.SPI\_FCR 174](#_Toc450425914)

[Table 332.SPI\_FWCR 174](#_Toc450425915)

[Table 333.SPI\_DLYCR 175](#_Toc450425916)

[Table 334.SPI\_TxCR 176](#_Toc450425917)

[Table 335.SPI\_RxCR 176](#_Toc450425918)

[Table 336.SPI\_SSCR 176](#_Toc450425919)

[Table 337.SPI\_ISR 177](#_Toc450425920)

[Table 338.SPI\_ FIFO\_STAT 178](#_Toc450425921)

[Table 339.SPI\_ TX\_REGx (x = 0, 1, 2, 3) 179](#_Toc450425922)

[Table 340.SPI\_ RX\_REGx (x = 0, 1, 2, 3) 179](#_Toc450425923)

[Table 341. SPI\_BOOT\_CR 179](#_Toc450425924)

[Table 342. SPI\_BOOT\_CR 179](#_Toc450425925)

[Table 343.IVA AccelerationRegister summary 183](#_Toc450425926)

[Table 344.Crypto Engine Register summary 226](#_Toc450425927)

[Table 345.spud\_irq\_en register 229](#_Toc450425928)

[Table 346.spdu\_irq\_stat register 229](#_Toc450425929)

[Table 347.dma\_brst\_sz register 229](#_Toc450425930)

[Table 348.sp\_decode register 229](#_Toc450425931)

[Table 349.spacc\_cmd\_priority register 229](#_Toc450425932)

[Table 350.sp\_version register 230](#_Toc450425933)

[Table 351.sp\_version\_ext register 230](#_Toc450425934)

[Table 352.spdu\_version register 230](#_Toc450425935)

[Table 353.RE register 230](#_Toc450425936)

[Table 354.MPM register 230](#_Toc450425937)

[Table 355.KEP register 230](#_Toc450425938)

[Table 356.EA register 230](#_Toc450425939)

[Table 357.RNG register 230](#_Toc450425940)

[Table 358.PKA register 231](#_Toc450425941)

[Table 359.SPACC register 231](#_Toc450425942)

[Table 360.IRQ\_EN register 231](#_Toc450425943)

[Table 361.IRQ\_STAT register 231](#_Toc450425944)

[Table 362.IRQ\_CTRL(SPAcc-QoS) register 231](#_Toc450425945)

[Table 363.IRQ\_CTRL(SPAcc) register 232](#_Toc450425946)

[Table 364. FIFO\_STAT(SPAcc-QoS) register 232](#_Toc450425947)

[Table 365.FIFO\_STAT(SPAcc) register 232](#_Toc450425948)

[Table 366.SDMA\_BRST\_SZ register 232](#_Toc450425949)

[Table 367.SRC\_PTR register 232](#_Toc450425950)

[Table 368.DST\_PTR register 233](#_Toc450425951)

[Table 369.OFFSET register 233](#_Toc450425952)

[Table 370.PRE\_ADD\_LEN register 233](#_Toc450425953)

[Table 371.POST\_ADD\_LEN register 233](#_Toc450425954)

[Table 372.PROC\_LEN register 233](#_Toc450425955)

[Table 373.ICV\_LEN register 233](#_Toc450425956)

[Table 374.ICV\_OFFSET register 233](#_Toc450425957)

[Table 375.IV\_OFFSET register 234](#_Toc450425958)

[Table 376.SW\_CTRL register 234](#_Toc450425959)

[Table 377.AUX\_INFO register 234](#_Toc450425960)

[Table 378.CTRL register 234](#_Toc450425961)

[Table 379.STAT\_POP register 235](#_Toc450425962)

[Table 380.STATUS register 235](#_Toc450425963)

[Table 381.STAT\_WD\_CTRL register 235](#_Toc450425964)

[Table 382.KEY\_SZ register 235](#_Toc450425965)

[Table 383.VSPACC\_RQST register 235](#_Toc450425966)

[Table 384.VSPACC\_ALLOC register 236](#_Toc450425967)

[Table 385.VSPACC\_PRIORITY register 236](#_Toc450425968)

[Table 386.VSPACC\_RC4\_KEY\_REQ register 236](#_Toc450425969)

[Table 387.VSPACC\_RC4\_KEY\_GNT register 236](#_Toc450425970)

[Table 388.VERSION register 236](#_Toc450425971)

[Table 389.VERSION\_EXT register 236](#_Toc450425972)

[Table 390.VERSION\_EXT\_2(SPAcc-QoS) register 237](#_Toc450425973)

[Table 391.VERSION\_EXT\_2(SPAcc) register 237](#_Toc450425974)

[Table 392.SECURE\_CTRL register 237](#_Toc450425975)

[Table 393.SECURE\_CTX\_RELEASE register 237](#_Toc450425976)

[Table 394.CIPH\_KEY register 238](#_Toc450425977)

[Table 395.HASH\_KEY register 238](#_Toc450425978)

[Table 396.RC4\_CTX register 238](#_Toc450425979)

[Table 397.Control register 238](#_Toc450425980)

[Table 398.Program Entry Address register 238](#_Toc450425981)

[Table 399.Return Code register 239](#_Toc450425982)

[Table 400.Build Configuration Register 239](#_Toc450425983)

[Table 401.Function Stack Pointer Register 240](#_Toc450425984)

[Table 402.Instructions Since Go register 240](#_Toc450425985)

[Table 403.Parameter Stack Pointer register 240](#_Toc450425986)

[Table 404.Configuration register 240](#_Toc450425987)

[Table 405.Status register 240](#_Toc450425988)

[Table 406.Flags register 240](#_Toc450425989)

[Table 407.Watchdog register 241](#_Toc450425990)

[Table 408.Cycles Since Go register 241](#_Toc450425991)

[Table 409.Index I register 241](#_Toc450425992)

[Table 410.Index J register 241](#_Toc450425993)

[Table 411.Index k register 241](#_Toc450425994)

[Table 412.Index L register 241](#_Toc450425995)

[Table 413.Interrupt Enable register 241](#_Toc450425996)

[Table 414.Jump Probability register 241](#_Toc450425997)

[Table 415.Probability LFSR Seed register 242](#_Toc450425998)

[Table 416.Bank Switch A register 242](#_Toc450425999)

[Table 417.Bank Switch B register 242](#_Toc450426000)

[Table 418.Bank Switch register 242](#_Toc450426001)

[Table 419.Bank Switch register 242](#_Toc450426002)

[Table 420.Control Register 242](#_Toc450426003)

[Table 421.TRNG\_IRQ\_STAT register 243](#_Toc450426004)

[Table 422.TRNG\_IRQ\_EN register 243](#_Toc450426005)

[Table 423.TRNG\_CFG register 243](#_Toc450426006)

[Table 424.TRNG DataA0 register 243](#_Toc450426007)

[Table 425.TRNG DataA1 register 243](#_Toc450426008)

[Table 426.TRNG DataA2 register 243](#_Toc450426009)

[Table 427.TRNG DataA3 register 243](#_Toc450426010)

[Table 428.DMA Engine Register summary 245](#_Toc450426011)

[Table 429.DMA manger status register (DSR) 246](#_Toc450426012)

[Table 430.DMA program counter register (DPC) 247](#_Toc450426013)

[Table 431.Interrupt enable register (INTEN) 247](#_Toc450426014)

[Table 432.Event-Interrupt raw status register (INT\_EVENT\_RIS) 247](#_Toc450426015)

[Table 433.Interrupt status register (INTMIS) 247](#_Toc450426016)

[Table 434.Interrupt clear register (INTCLR) 248](#_Toc450426017)

[Table 435.Fault status DMA manger register (FSRD) 248](#_Toc450426018)

[Table 436.Fault status DMA channel register (FSRC) 248](#_Toc450426019)

[Table 437.Fault type DMA manager register (FTRD) 248](#_Toc450426020)

[Table 438.Fault type DMA channel register (FTRn) 249](#_Toc450426021)

[Table 439.Channel status registers (CSRn) 251](#_Toc450426022)

[Table 440.Channel program counter register (CPCn) 251](#_Toc450426023)

[Table 441.Destination address registers(DARn) 252](#_Toc450426024)

[Table 442.Channel control register(CCRn) 252](#_Toc450426025)

[Table 443.Loop counter 0 registers (LC0\_n) 254](#_Toc450426026)

[Table 444.Loop counter 1 registers (LC1\_n) 254](#_Toc450426027)

[Table 445.Debug status register(DBGSTATUS) 254](#_Toc450426028)

[Table 446.Debug command register (DBGCMD) 254](#_Toc450426029)

[Table 447.Debug instruction-0 register(DBGINST0) 254](#_Toc450426030)

[Table 448.Debug instruction-1 register(DBGINST1) 255](#_Toc450426031)

[Table 449.Configuration register 0 (CR0) 255](#_Toc450426032)

[Table 450.Configuration register 1 (CR1) 256](#_Toc450426033)

[Table 451.Configuration register 2 (CR2) 256](#_Toc450426034)

[Table 452.Configuration register 3 (CR3) 256](#_Toc450426035)

[Table 453.Configuration register 4 (CR4) 256](#_Toc450426036)

[Table 454.DMA configuration register (CRD) 256](#_Toc450426037)

[Table 455.Watchdog register(WD) 257](#_Toc450426038)

[Table 456.Peripheral identification registers (periph\_id\_0) 257](#_Toc450426039)

[Table 457.Peripheral identification registers (periph\_id\_1) 258](#_Toc450426040)

[Table 458.Peripheral identification registers (periph\_id\_2) 258](#_Toc450426041)

[Table 459.Peripheral identification registers (periph\_id\_3) 258](#_Toc450426042)

[Table 460.Instruction syntax summary 262](#_Toc450426043)

[Table 461.GMAC Controller Register summary 275](#_Toc450426044)

[Table 462.Bus mode register CSR0 275](#_Toc450426045)

[Table 463.Transmit automatic polling intervals 276](#_Toc450426046)

[Table 464.The transmit poll demand register CSR1 276](#_Toc450426047)

[Table 465.The receive poll demand register CSR2 277](#_Toc450426048)

[Table 466.The receive descriptor list base address register CSR3 277](#_Toc450426049)

[Table 467.The transmit descriptor list base address register CSR4 277](#_Toc450426050)

[Table 468.The status register CSR5 277](#_Toc450426051)

[Table 469.The operation mode register CSR6 279](#_Toc450426052)

[Table 470.The interrupt enable register CSR7 280](#_Toc450426053)

[Table 471.The missed frames counter register CSR8 281](#_Toc450426054)

[Table 472.The serial ROM / Software MII Serial Management register CSR9 281](#_Toc450426055)

[Table 473.The MII serial management register CSR10 282](#_Toc450426056)

[Table 474.The general-purpose timer and interrupt mitigation control register CSR11 283](#_Toc450426057)

[Table 475.MAC address low register CSR16 283](#_Toc450426058)

[Table 476.MAC MAC address high register CSR17 284](#_Toc450426059)

[Table 477.Pause time & cache thresholds register CSR18 284](#_Toc450426060)

[Table 478.FIFO thresholds register CSR19 284](#_Toc450426061)

[Table 479.Flow control setup & status CSR20 285](#_Toc450426062)

[Table 480.SDIO Controller Register summary 287](#_Toc450426063)

[Table 481.SDIO\_HRS0 register 288](#_Toc450426064)

[Table 482.SDIO\_HRS01 Register 288](#_Toc450426065)

[Table 483.SDIO\_HRS02 Register 289](#_Toc450426066)

[Table 484.SDIO\_HRS04 Register 289](#_Toc450426067)

[Table 485.SDIO\_HRS05 Register 289](#_Toc450426068)

[Table 486.SDIO\_HRS06 Register 289](#_Toc450426069)

[Table 487.SDIO\_HRS08 Register 289](#_Toc450426070)

[Table 488.SDIO\_HRS09 Register 290](#_Toc450426071)

[Table 489.SDIO\_HRS10 Register 290](#_Toc450426072)

[Table 490.SDIO\_HRS12 Register 290](#_Toc450426073)

[Table 491.SDIO\_HRS13 Register 290](#_Toc450426074)

[Table 492.SDIO\_HRS14 Register 290](#_Toc450426075)

[Table 493.SDIO\_HRS20 Register 290](#_Toc450426076)

[Table 494.SDIO\_HRS Register Fields 291](#_Toc450426077)

[Table 495.SDIO\_HRS24 Register 291](#_Toc450426078)

[Table 496.SDIO\_HRS25 Register 291](#_Toc450426079)

[Table 497.SDIO\_HRS26 Register 292](#_Toc450426080)

[Table 498.SDIO\_HRS27 Register 292](#_Toc450426081)

[Table 499.SDIO\_HRS28 Register 292](#_Toc450426082)

[Table 500.SDIO\_HRS29 Register 292](#_Toc450426083)

[Table 501.SDIO\_HRS30 Register 292](#_Toc450426084)

[Table 502.SDIO\_HRS31 Register 292](#_Toc450426085)

[Table 503.SDIO\_HRS32 Register 292](#_Toc450426086)

[Table 504.SDIO\_HRS33 Register 292](#_Toc450426087)

[Table 505.SDIO\_HRS34 Register 293](#_Toc450426088)

[Table 506.SDIO\_HRS35 Register 293](#_Toc450426089)

[Table 507.SDIO\_SRS0 Register 293](#_Toc450426090)

[Table 508.SDIO\_SRS1 Register 293](#_Toc450426091)

[Table 509.SDIO\_SRS2 Register 294](#_Toc450426092)

[Table 510.SDIO\_SRS3 Register 294](#_Toc450426093)

[Table 511.SDIO\_SRS4 Register 297](#_Toc450426094)

[Table 512.SDIO\_SRS5 Register 297](#_Toc450426095)

[Table 513.SDIO\_SRS6 Register 297](#_Toc450426096)

[Table 514.SDIO\_SRS7 Register 297](#_Toc450426097)

[Table 515.SDIO\_SRS8 Register 297](#_Toc450426098)

[Table 516.SDIO\_SRS9 Register 298](#_Toc450426099)

[Table 517.SDIO\_SRS10 Register 301](#_Toc450426100)

[Table 518.SDIO\_SRS11 Register 303](#_Toc450426101)

[Table 519.SDIO\_SRS12 Register 306](#_Toc450426102)

[Table 520.SDIO\_SRS13 Register 309](#_Toc450426103)

[Table 521.SDIO\_SRS14 Register 310](#_Toc450426104)

[Table 522.SDIO\_SRS15 Register 312](#_Toc450426105)

[Table 523.SDIO\_SRS16 Register 314](#_Toc450426106)

[Table 524.SDIO\_SRS17 Register 316](#_Toc450426107)

[Table 525.SDIO\_SRS18 Register 317](#_Toc450426108)

[Table 526.Event Trigger Register Fields 317](#_Toc450426109)

[Table 527.SDIO\_SRS21 Register 318](#_Toc450426110)

[Table 528.ADMA System Address Register Fields 318](#_Toc450426111)

[Table 529.SDIO\_SRS24 Register 319](#_Toc450426112)

[Table 530.SDIO\_SRS25 Register 319](#_Toc450426113)

[Table 531.SDIO\_SRS26 Register 319](#_Toc450426114)

[Table 532.SDIO\_SRS27 Register 320](#_Toc450426115)

[Table 533.SDIO\_SRS56 Register 320](#_Toc450426116)

[Table 534.CRS63 Register 321](#_Toc450426117)

[Table 535.Card Initialization chart 322](#_Toc450426118)

[Table 536.DDR Controller&PHY Register summary 332](#_Toc450426119)

[Table 537.DENALI\_CTL\_00 register 338](#_Toc450426120)

[Table 538.DENALI\_CTL\_01 register 338](#_Toc450426121)

[Table 539.DENALI\_CTL\_02 register 339](#_Toc450426122)

[Table 540.DENALI\_CTL\_03register 339](#_Toc450426123)

[Table 541.DENALI\_CTL\_04egister 339](#_Toc450426124)

[Table 542.DENALI\_CTL\_05 register 339](#_Toc450426125)

[Table 543.DENALI\_CTL\_06 register 340](#_Toc450426126)

[Table 544.DENALI\_CTL\_07register 340](#_Toc450426127)

[Table 545.DENALI\_CTL\_08 register 340](#_Toc450426128)

[Table 546.DENALI\_CTL\_09register 340](#_Toc450426129)

[Table 547.DENALI\_CTL\_10register 340](#_Toc450426130)

[Table 548.DENALI\_CTL\_11 register 340](#_Toc450426131)

[Table 549.DENALI\_CTL\_12register 340](#_Toc450426132)

[Table 550.DENALI\_CTL\_13 register 341](#_Toc450426133)

[Table 551.DENALI\_CTL\_14register 341](#_Toc450426134)

[Table 552.DENALI\_CTL\_15register 341](#_Toc450426135)

[Table 553.DENALI\_CTL\_16 register 341](#_Toc450426136)

[Table 554.DENALI\_CTL\_17 register 341](#_Toc450426137)

[Table 555.DENALI\_CTL\_18 register 342](#_Toc450426138)

[Table 556.DENALI\_CTL\_19 register 342](#_Toc450426139)

[Table 557.DENALI\_CTL\_20 register 342](#_Toc450426140)

[Table 558.DENALI\_CTL\_21 register 342](#_Toc450426141)

[Table 559.DENALI\_CTL\_22 register 342](#_Toc450426142)

[Table 560.DENALI\_CTL\_23 register 343](#_Toc450426143)

[Table 561.DENALI\_CTL\_24 register 343](#_Toc450426144)

[Table 562.DENALI\_CTL\_25 register 343](#_Toc450426145)

[Table 563.DENALI\_CTL\_26 register 343](#_Toc450426146)

[Table 564.DENALI\_CTL\_27 register 343](#_Toc450426147)

[Table 565.DENALI\_CTL\_28 register 344](#_Toc450426148)

[Table 566.DENALI\_CTL\_29 register 344](#_Toc450426149)

[Table 567.DENALI\_CTL\_30 register 344](#_Toc450426150)

[Table 568.DENALI\_CTL\_31 register 344](#_Toc450426151)

[Table 569.DENALI\_CTL\_32 register 345](#_Toc450426152)

[Table 570.DENALI\_CTL\_33 register 345](#_Toc450426153)

[Table 571.DENALI\_CTL\_34 register 345](#_Toc450426154)

[Table 572.DENALI\_CTL\_35 register 345](#_Toc450426155)

[Table 573.DENALI\_CTL\_36 register 346](#_Toc450426156)

[Table 574.DENALI\_CTL\_37 register 346](#_Toc450426157)

[Table 575.DENALI\_CTL\_38 register 346](#_Toc450426158)

[Table 576.DENALI\_CTL\_39 register 346](#_Toc450426159)

[Table 577.DENALI\_CTL\_40 register 346](#_Toc450426160)

[Table 578.DENALI\_CTL\_41 register 347](#_Toc450426161)

[Table 579.DENALI\_CTL\_42 register 347](#_Toc450426162)

[Table 580.DENALI\_CTL\_43 register 347](#_Toc450426163)

[Table 581.DENALI\_CTL\_44 register 347](#_Toc450426164)

[Table 582.DENALI\_CTL\_45 register 347](#_Toc450426165)

[Table 583.DENALI\_CTL\_46 register 348](#_Toc450426166)

[Table 584.DENALI\_CTL\_47 register 348](#_Toc450426167)

[Table 585.DENALI\_CTL\_48 register 348](#_Toc450426168)

[Table 586.DENALI\_CTL\_49 register 348](#_Toc450426169)

[Table 587.DENALI\_CTL\_50 register 348](#_Toc450426170)

[Table 588.DENALI\_CTL\_51 register 348](#_Toc450426171)

[Table 589.DENALI\_CTL\_52 register 349](#_Toc450426172)

[Table 590.DENALI\_CTL\_53 register 349](#_Toc450426173)

[Table 591.DENALI\_CTL\_54 register 349](#_Toc450426174)

[Table 592.DENALI\_CTL\_55 register 349](#_Toc450426175)

[Table 593.DENALI\_CTL\_56 register 349](#_Toc450426176)

[Table 594.DENALI\_CTL\_57 register 349](#_Toc450426177)

[Table 595.DENALI\_CTL\_58 register 350](#_Toc450426178)

[Table 596.DENALI\_CTL\_59 register 350](#_Toc450426179)

[Table 597.DENALI\_CTL\_60 register 350](#_Toc450426180)

[Table 598.DENALI\_CTL\_61 register 350](#_Toc450426181)

[Table 599.DENALI\_CTL\_62 register 351](#_Toc450426182)

[Table 600.DENALI\_CTL\_63 register 351](#_Toc450426183)

[Table 601.DENALI\_CTL\_64 register 351](#_Toc450426184)

[Table 602.DENALI\_CTL\_65 register 352](#_Toc450426185)

[Table 603.DENALI\_CTL\_66 register 352](#_Toc450426186)

[Table 604.DENALI\_CTL\_67 register 352](#_Toc450426187)

[Table 605.DENALI\_CTL\_68 register 353](#_Toc450426188)

[Table 606.DENALI\_CTL\_69 register 353](#_Toc450426189)

[Table 607.DENALI\_CTL\_70 register 353](#_Toc450426190)

[Table 608.DENALI\_CTL\_71 register 353](#_Toc450426191)

[Table 609.DENALI\_CTL\_72 register 353](#_Toc450426192)

[Table 610.DENALI\_CTL\_73 register 354](#_Toc450426193)

[Table 611.DENALI\_CTL\_74 register 354](#_Toc450426194)

[Table 612.DENALI\_CTL\_75 register 354](#_Toc450426195)

[Table 613.DENALI\_CTL\_76 register 354](#_Toc450426196)

[Table 614.DENALI\_CTL\_77 register 354](#_Toc450426197)

[Table 615.DENALI\_CTL\_78 register 354](#_Toc450426198)

[Table 616.DENALI\_CTL\_79 register 354](#_Toc450426199)

[Table 617.DENALI\_CTL\_80 register 355](#_Toc450426200)

[Table 618.DENALI\_CTL\_81 register 355](#_Toc450426201)

[Table 619.DENALI\_CTL\_82 register 355](#_Toc450426202)

[Table 620.DENALI\_CTL\_83 register 355](#_Toc450426203)

[Table 621.DENALI\_CTL\_84 register 355](#_Toc450426204)

[Table 622.DENALI\_CTL\_85 register 355](#_Toc450426205)

[Table 623.DENALI\_CTL\_86 register 355](#_Toc450426206)

[Table 624.DENALI\_CTL\_87 register 355](#_Toc450426207)

[Table 625.DENALI\_CTL\_88 register 356](#_Toc450426208)

[Table 626.DENALI\_CTL\_89 register 356](#_Toc450426209)

[Table 627.DENALI\_CTL\_90 register 356](#_Toc450426210)

[Table 628.DENALI\_CTL\_91 register 356](#_Toc450426211)

[Table 629.DENALI\_CTL\_92 register 356](#_Toc450426212)

[Table 630.DENALI\_CTL\_93 register 357](#_Toc450426213)

[Table 631.DENALI\_CTL\_94 register 357](#_Toc450426214)

[Table 632.DENALI\_CTL\_95 register 357](#_Toc450426215)

[Table 633.DENALI\_CTL\_96 register 357](#_Toc450426216)

[Table 634.DENALI\_CTL\_97 register 358](#_Toc450426217)

[Table 635.DENALI\_CTL\_98 register 358](#_Toc450426218)

[Table 636.DENALI\_CTL\_99 register 358](#_Toc450426219)

[Table 637.DENALI\_CTL\_100 register 358](#_Toc450426220)

[Table 638.DENALI\_CTL\_101 register 359](#_Toc450426221)

[Table 639.DENALI\_CTL\_102 register 359](#_Toc450426222)

[Table 640.DENALI\_CTL\_103 register 359](#_Toc450426223)

[Table 641.DENALI\_CTL\_104 register 359](#_Toc450426224)

[Table 642.DENALI\_CTL\_105 register 360](#_Toc450426225)

[Table 643.DENALI\_CTL\_106 register 360](#_Toc450426226)

[Table 644.DENALI\_CTL\_107 register 360](#_Toc450426227)

[Table 645.DENALI\_CTL\_109 register 360](#_Toc450426228)

[Table 646.DENALI\_CTL\_110 register 360](#_Toc450426229)

[Table 647.DENALI\_CTL\_111 register 361](#_Toc450426230)

[Table 648.DENALI\_CTL\_112 register 361](#_Toc450426231)

[Table 649.DENALI\_CTL\_113 register 361](#_Toc450426232)

[Table 650.DENALI\_CTL\_114 register 361](#_Toc450426233)

[Table 651.DENALI\_CTL\_115 register 362](#_Toc450426234)

[Table 652.DENALI\_CTL\_116 register 362](#_Toc450426235)

[Table 653.DENALI\_CTL\_117 register 362](#_Toc450426236)

[Table 654.DENALI\_CTL\_118 register 362](#_Toc450426237)

[Table 655.DENALI\_CTL\_119 register 363](#_Toc450426238)

[Table 656.DENALI\_CTL\_120 register 363](#_Toc450426239)

[Table 657.DENALI\_CTL\_121 register 363](#_Toc450426240)

[Table 658.DENALI\_CTL\_122 register 363](#_Toc450426241)

[Table 659.DENALI\_CTL\_123 register 363](#_Toc450426242)

[Table 660.DENALI\_CTL\_124 register 364](#_Toc450426243)

[Table 661.DENALI\_CTL\_125 register 364](#_Toc450426244)

[Table 662.DENALI\_CTL\_126 register 364](#_Toc450426245)

[Table 663.DENALI\_CTL\_127 register 364](#_Toc450426246)

[Table 664.DENALI\_CTL\_128 register 365](#_Toc450426247)

[Table 665.DENALI\_CTL\_129 register 365](#_Toc450426248)

[Table 666.DENALI\_CTL\_131 register 365](#_Toc450426249)

[Table 667.DENALI\_CTL\_132 register 365](#_Toc450426250)

[Table 668.DENALI\_CTL\_133 register 366](#_Toc450426251)

[Table 669.DENALI\_CTL\_134 register 366](#_Toc450426252)

[Table 670.DENALI\_CTL\_135 register 366](#_Toc450426253)

[Table 671.DENALI\_CTL\_136 register 366](#_Toc450426254)

[Table 672.DENALI\_CTL\_137 register 366](#_Toc450426255)

[Table 673.DENALI\_CTL\_138 register 367](#_Toc450426256)

[Table 674.DENALI\_CTL\_139 register 367](#_Toc450426257)

[Table 675.DENALI\_CTL\_140 register 367](#_Toc450426258)

[Table 676.DENALI\_CTL\_141 register 367](#_Toc450426259)

[Table 677.DENALI\_CTL\_142 register 367](#_Toc450426260)

[Table 678.DENALI\_CTL\_143 register 367](#_Toc450426261)

[Table 679.DENALI\_CTL\_144 register 368](#_Toc450426262)

[Table 680. Program Register 368](#_Toc450426263)

[Table 681.I2C Controller Register summary 379](#_Toc450426264)

[Table 682.I2C\_MTXR 379](#_Toc450426265)

[Table 683.I2C\_MRXR 379](#_Toc450426266)

[Table 684.I2C\_STXR 379](#_Toc450426267)

[Table 685.I2C\_SRXR 379](#_Toc450426268)

[Table 686.I2C\_SADDR 379](#_Toc450426269)

[Table 687.I2C\_IER 380](#_Toc450426270)

[Table 688.I2C\_ISR 380](#_Toc450426271)

[Table 689.I2C\_LCMR 381](#_Toc450426272)

[Table 690.I2C\_LSR 381](#_Toc450426273)

[Table 691.I2C\_CONR 381](#_Toc450426274)

[Table 692.I2C\_OPR 382](#_Toc450426275)

[Table 693.SPI Controller Register summary 390](#_Toc450426276)

[Table 694.SPI\_TxR 390](#_Toc450426277)

[Table 695.SPI\_RxR 390](#_Toc450426278)

[Table 696.SPI\_IER 390](#_Toc450426279)

[Table 697.SPI\_FCR 392](#_Toc450426280)

[Table 698.SPI\_FWCR 392](#_Toc450426281)

[Table 699.SPI\_DLYCR 393](#_Toc450426282)

[Table 700.SPI\_TxCR 394](#_Toc450426283)

[Table 701.SPI\_RxCR 394](#_Toc450426284)

[Table 702.SPI\_SSCR 394](#_Toc450426285)

[Table 703.SPI\_ISR 395](#_Toc450426286)

[Table 704.SPI\_ FIFO\_STAT 396](#_Toc450426287)

[Table 705.SPI\_ TX\_REGx (x = 0, 1, 2, 3) 397](#_Toc450426288)

[Table 706.SPI\_ RX\_REGx (x = 0, 1, 2, 3) 397](#_Toc450426289)

[Table 707.Hardware command 397](#_Toc450426290)

[Table 708.UART Controller Register summary 401](#_Toc450426291)

[Table 709.UART\_RBR 401](#_Toc450426292)

[Table 710.UART\_THR 401](#_Toc450426293)

[Table 711.UART\_IER 401](#_Toc450426294)

[Table 712.UART\_IIR 402](#_Toc450426295)

[Table 713.UART\_FCR 402](#_Toc450426296)

[Table 714.UART\_LCR 403](#_Toc450426297)

[Table 715.UART\_MCR 403](#_Toc450426298)

[Table 716.UART\_LSR 404](#_Toc450426299)

[Table 717.UART\_MSR 405](#_Toc450426300)

[Table 718.UART\_ICR 405](#_Toc450426301)

[Table 719.UART\_CTRLR 405](#_Toc450426302)

[Table 720.UART\_MODE 405](#_Toc450426303)

[Table 721.UART\_DLL 406](#_Toc450426304)

[Table 722.UART\_DLH 406](#_Toc450426305)

[Table 723.Watch Timer Register summary 409](#_Toc450426306)

[Table 724.WDTLR 409](#_Toc450426307)

[Table 725.WDTCVR 409](#_Toc450426308)

[Table 726.WDTCON 409](#_Toc450426309)

[Table 727.TEST 409](#_Toc450426310)

[Table 728.Timer Register summary 411](#_Toc450426311)

[Table 729.TMRxLR(x=0,1,2) 411](#_Toc450426312)

[Table 730.TMRxCVR(x=0,1,2) 411](#_Toc450426313)

[Table 731.TMRxCON(x=0,1,2) 411](#_Toc450426314)

[Table 732.TMRMODE 412](#_Toc450426315)

[Table 733.GPIO Controller Register summary 413](#_Toc450426316)

[Table 734.GPIO\_PxDR(x=A,B,C,D) 413](#_Toc450426317)

[Table 735.GPIO\_PxCON(x=A,B,C,D) 413](#_Toc450426318)

[Table 736.GPIO\_TEST 414](#_Toc450426319)

[Table 737.GPIO\_IEx(x=A,B,C,D) 414](#_Toc450426320)

[Table 738.GPIO\_ISx(x=A,B,C,D) 414](#_Toc450426321)

[Table 739.GPIO\_IBEx(x=A,B,C,D) 414](#_Toc450426322)

[Table 740.GPIO\_IEVx(x=A,B,C,D) 414](#_Toc450426323)

[Table 741.GPIO\_ICx(x=A,B,C,D) 415](#_Toc450426324)

[Table 742.GPIO\_ISRx(x=A,B,C,D) 415](#_Toc450426325)

[Table 743.Rising edge detected timing 416](#_Toc450426326)

[Table 744.PWMRegister summary 419](#_Toc450426327)

[Table 745.PWMTx\_CNTR(x=0~3) 419](#_Toc450426328)

[Table 746.PWMTx\_HRC(x=0~3) 419](#_Toc450426329)

[Table 747.PWMTx\_LRC(x=0~3) 419](#_Toc450426330)

[Table 748.PWMTx\_CTRL(x=0~3) 419](#_Toc450426331)

[Table 749.IOMUX Register summary 421](#_Toc450426332)

[Table 750.PAD\_CTRL\_0 Register 424](#_Toc450426333)

[Table 751.PAD\_CTRL\_1 Register 425](#_Toc450426334)

[Table 752.PAD\_CTRL\_2 Register 425](#_Toc450426335)

[Table 753.PAD\_CTRL\_3 Register 426](#_Toc450426336)

[Table 754.PAD\_CTRL\_4 Register 427](#_Toc450426337)

[Table 755.PAD\_CTRL\_5 Register 428](#_Toc450426338)

[Table 756.PAD\_CTRL\_6 Register 429](#_Toc450426339)

[Table 757.PAD\_CTRL\_7 Register 431](#_Toc450426340)

[Table 758.PAD\_CTRL\_8 Register 432](#_Toc450426341)

[Table 759.PAD\_CTRL\_9 Register 433](#_Toc450426342)

[Table 760.PAD\_CTRL\_10 Register 434](#_Toc450426343)

[Table 761.PAD\_CTRL\_11 Register 435](#_Toc450426344)

[Table 762.PAD\_CTRL\_12 Register 437](#_Toc450426345)

[Table 763.PAD\_CTRL\_13 Register 438](#_Toc450426346)

[Table 764.PAD\_CTRL\_14 Register 439](#_Toc450426347)

[Table 765.PAD\_CTRL\_15 Register 440](#_Toc450426348)

[Table 766.PAD\_CTRL\_16 Register 442](#_Toc450426349)

[Table 767.PAD\_CTRL\_17 Register 442](#_Toc450426350)

[Table 768.PAD\_CTRL\_18 Register 443](#_Toc450426351)

[Table 769.PAD\_CTRL\_19 Register 445](#_Toc450426352)

[Table 770.PAD\_CTRL\_20 Register 446](#_Toc450426353)

[Table 771.PAD\_CTRL\_21 Register 447](#_Toc450426354)

[Table 772.PAD\_CTRL\_22 Register 448](#_Toc450426355)

[Table 773.PAD\_CTRL\_23 Register 450](#_Toc450426356)

[Table 774.PAD\_CTRL\_24 Register 451](#_Toc450426357)

[Table 775.PAD\_CTRL\_25 Register 453](#_Toc450426358)

[Table 776.PAD\_CTRL\_26 Register 454](#_Toc450426359)

[Table 777.PAD\_CTRL\_27 Register 455](#_Toc450426360)

[Table 778.PAD\_CTRL\_28 Register 457](#_Toc450426361)

[Table 779.PAD\_CTRL\_29 Register 457](#_Toc450426362)

[Table 780.PAD\_CTRL\_30 Register 458](#_Toc450426363)

[Table 781.PAD\_CTRL\_31 Register 459](#_Toc450426364)

[Table 782.PAD\_CTRL\_32 Register 461](#_Toc450426365)

[Table 783.PAD\_CTRL\_33 Register 462](#_Toc450426366)

[Table 784.PAD\_CTRL\_34 Register 463](#_Toc450426367)

[Table 785.PAD\_CTRL\_35 Register 464](#_Toc450426368)

[Table 786.PAD\_CTRL\_36 Register 465](#_Toc450426369)

[Table 787.PAD\_CTRL\_35 Register 466](#_Toc450426370)

[Table 788.AD\_CTRL\_38 Register 468](#_Toc450426371)

[Table 789.PAD\_CTRL\_39 Register 469](#_Toc450426372)

[Table 790.PAD\_CTRL\_40 Register 470](#_Toc450426373)

[Table 791.PAD\_CTRL\_41 Register 471](#_Toc450426374)

[Table 792.SW\_MUX\_SEL Register 472](#_Toc450426375)

[Table 793.DDR\_DBG\_SEL Register 473](#_Toc450426376)

[Table 794.SDIO\_DBG\_SEL Register 473](#_Toc450426377)

[Table 795.DMA\_DBG\_SEL Register 473](#_Toc450426378)

[Table 796.ISRAM\_DBG\_SEL Register 473](#_Toc450426379)

[Table 797.Normal & Engineering & FPGA Mode IOMUX Table 474](#_Toc450426380)

[Table 798.DFT & MBIST Mode IOMUX Table 476](#_Toc450426381)

[Table 799.Analog Test Mode IOMUX Table 479](#_Toc450426382)

[Table 800.NAND-Tree Test Mode IOMUX Table 482](#_Toc450426383)

[Table 801.Debug Mode MUX Table 484](#_Toc450426384)

[Table 802.Programming configuration 489](#_Toc450426385)

List of Figures

[Figure 1.F2 Block Diagram 26](#_Toc450426388)

[Figure 2.Transmit programming 113](#_Toc450426389)

[Figure 3 Coresight Debug mode 170](#_Toc450426390)

[Figure 4. Coresight Trace mode 171](#_Toc450426391)

[Figure 5.SPI clock’s delay meaning 176](#_Toc450426392)

[Figure 6. APB write flow 180](#_Toc450426393)

[Figure 7. APB read data 181](#_Toc450426394)

[Figure 8. AHB read flow 182](#_Toc450426395)

[Figure 9.IVA diagram block 225](#_Toc450426396)

[Figure 10.Crypto Engine Programming Sequence 244](#_Toc450426397)

[Figure 11.Thread operating states 259](#_Toc450426398)

[Figure 12.DMAADDH encoding 263](#_Toc450426399)

[Figure 13.DMAADNH encoding 264](#_Toc450426400)

[Figure 14.DMAEND encoding 264](#_Toc450426401)

[Figure 15.DMAFLUSHP encoding 264](#_Toc450426402)

[Figure 16.DMAGO encoding 265](#_Toc450426403)

[Figure 17.DMAKILL encoding 266](#_Toc450426404)

[Figure 18.DMALD encoding 266](#_Toc450426405)

[Figure 19. DMALDP[S|B] encoding 267](#_Toc450426406)

[Figure 20.DMALP encoding 268](#_Toc450426407)

[Figure 21.DMALPEND encoding 268](#_Toc450426408)

[Figure 22.DMAMOV encoding 270](#_Toc450426409)

[Figure 23.DMANOP encoding 270](#_Toc450426410)

[Figure 24.DMARMB encoding 270](#_Toc450426411)

[Figure 25.DMASEV encoding 271](#_Toc450426412)

[Figure 26.DMAST encoding 271](#_Toc450426413)

[Figure 27.DMASTP encoding 272](#_Toc450426414)

[Figure 28.DMASTZ encoding 273](#_Toc450426415)

[Figure 29.DMAWFE encoding 273](#_Toc450426416)

[Figure 30.DMAWFP encoding 273](#_Toc450426417)

[Figure 31.DMAWMB encoding 274](#_Toc450426418)

[Figure 32.Transmit programming 285](#_Toc450426419)

[Figure 33.Receive programming 286](#_Toc450426420)

[Figure 34.Voltage Switching chart 324](#_Toc450426421)

[Figure 35.Clock Tuning chart 325](#_Toc450426422)

[Figure 36.Commands Which Do Not Use DAT Line 326](#_Toc450426423)

[Figure 37.Commands with BUSY 328](#_Toc450426424)

[Figure 38.Commands Which Transfer Data without DMA 330](#_Toc450426425)

[Figure 39.Write/Read Data Blocks Sub-procedure 331](#_Toc450426426)

[Figure 40.Commands Which Transfer Data Using DMA 332](#_Toc450426427)

[Figure 41. Initial programming 374](#_Toc450426428)

[Figure 42.MBIST programming 375](#_Toc450426429)

[Figure 43.Initial program 376](#_Toc450426430)

[Figure 44.Loopback programming 377](#_Toc450426431)

[Figure 45.Master/Transmitter Mode Operations 386](#_Toc450426432)

[Figure 46.Master/Receiver Mode Operations 387](#_Toc450426433)

[Figure 47.Slave/Transmitter Mode Operations 388](#_Toc450426434)

[Figure 48.Slave/Transmitter Mode Operations 389](#_Toc450426435)

[Figure 49.SPI clock’s delay meaning 394](#_Toc450426436)

[Figure 50.Command 0x01 397](#_Toc450426437)

[Figure 51.Command 0x02 397](#_Toc450426438)

[Figure 52.Command 0x03 ~ Command 0x06 398](#_Toc450426439)

[Figure 53.Command 0x08 398](#_Toc450426440)

[Figure 54.Command 0x09 ~ Command 0x0C 398](#_Toc450426441)

[Figure 55.SPI master controller transaction 399](#_Toc450426442)

[Figure 56.SPI slave controller transaction 400](#_Toc450426443)

[Figure 57.Transmit programming 408](#_Toc450426444)

[Figure 58.WDT control flow chart 410](#_Toc450426445)

[Figure 59.Falling edge detected timing 416](#_Toc450426446)

[Figure 60.Both-edges detected timing 417](#_Toc450426447)

[Figure 61.High-level detected timing 417](#_Toc450426448)

[Figure 62.High-level detected timing 418](#_Toc450426449)

[Figure 63.Bit map for main-clock pad control register 423](#_Toc450426450)

[Figure 64.PAD\_CTRL\_0 Register 424](#_Toc450426451)

[Figure 65.Bit map for SDIO pad control registers 424](#_Toc450426452)

[Figure 66.PAD\_CTRL\_1 Register 425](#_Toc450426453)

[Figure 67.Bit map for GMAC pad control registers 428](#_Toc450426454)

[Figure 68.PAD\_CTRL\_5 Register 428](#_Toc450426455)

[Figure 69.Bit map for dedicated GPIO pad control registers 437](#_Toc450426456)

[Figure 70.PAD\_CTRL\_13 Register 438](#_Toc450426457)

[Figure 71.Bit map for alternated GPIO pad control registers 450](#_Toc450426458)

[Figure 72.PAD\_CTRL\_23 Register 450](#_Toc450426459)

[Figure 73.Normal mode output timing 489](#_Toc450426460)

[Figure 74.Engineering mode output timing 489](#_Toc450426461)

[Figure 75.DDR\_DFI debug mode output timing 490](#_Toc450426462)

1. SOC Programming Overview
   1. Programming Block Diagram



Figure .F2 Block Diagram

* 1. Memory Map
     1. System Address Map

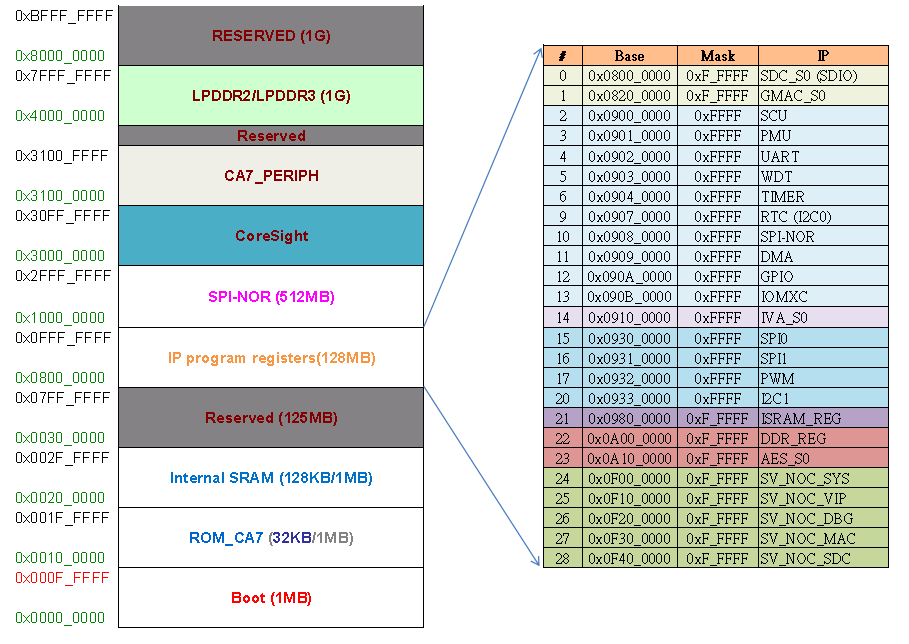


Table .System memory map

* + 1. Device Specific Memory Map

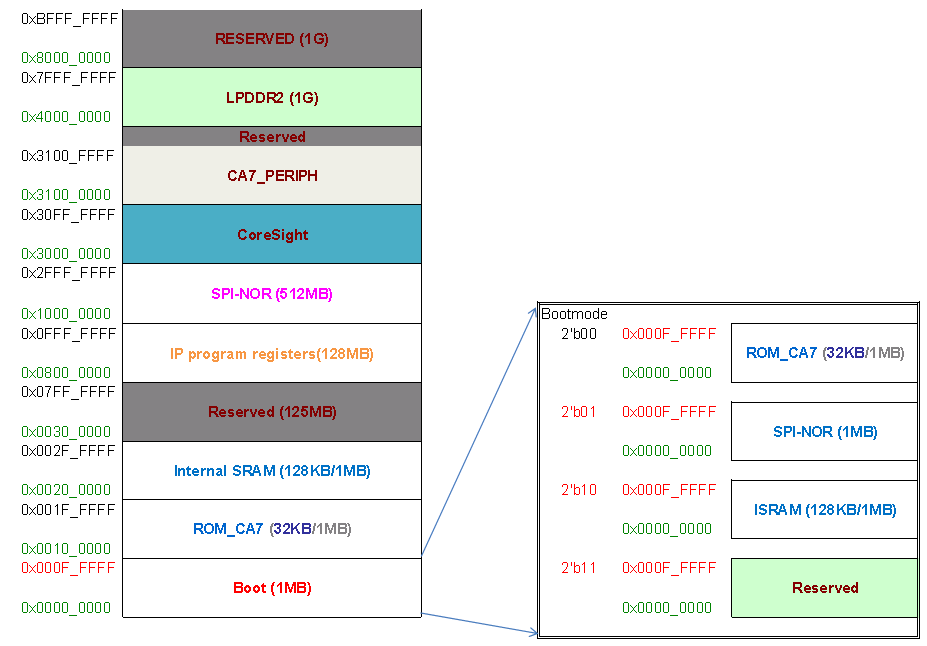


Table .Device memory map

* + 1. Specific Function Register for IO Devices

Table .Register for IO devices

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Base** | **Mask** | **IP** |
| 0 | 0x0800\_0000 | 0xF\_FFFF | SDC\_S0 (SDIO) |
| 1 | 0x0820\_0000 | 0xF\_FFFF | GMAC\_S0 |
| 2 | 0x0900\_0000 | 0xFFFF | SCU |
| 3 | 0x0901\_0000 | 0xFFFF | PMU |
| 4 | 0x0902\_0000 | 0xFFFF | UART |
| 5 | 0x0903\_0000 | 0xFFFF | WDT |
| 6 | 0x0904\_0000 | 0xFFFF | TIMER0 |
| 7 | 0x0905\_0000 | 0xFFFF | TIMER1 |
| 8 | 0x0906\_0000 | 0xFFFF | TIMER2 |
| 9 | 0x0907\_0000 | 0xFFFF | RTC (I2C0) |
| 10 | 0x0908\_0000 | 0xFFFF | SPI-NOR |
| 11 | 0x0909\_0000 | 0xFFFF | DMA |
| 12 | 0x090A\_0000 | 0xFFFF | GPIO |
| 13 | 0x090B\_0000 | 0xFFFF | IOMXC |
| 14 | 0x0910\_0000 | 0xFFFF | IVA\_S0 |
| 15 | 0x0930\_0000 | 0xFFFF | SPI0 |
| 16 | 0x0931\_0000 | 0xFFFF | SPI1 |
| 17 | 0x0932\_0000 | 0xFFFF | PWM0 |
| 18 | 0x0933\_0000 | 0xFFFF | PWM1 |
| 19 | 0x0934\_0000 | 0xFFFF | PWM2 |
| 20 | 0x0935\_0000 | 0xFFFF | I2C1 |
| 21 | 0x0980\_0000 | 0xF\_FFFF | ISRAM\_REG |
| 22 | 0x0A00\_0000 | 0xF\_FFFF | DDR\_REG |
| 23 | 0x0A10\_0000 | 0xF\_FFFF | AES\_S0 |
| 24 | 0x0F00\_0000 | 0xF\_FFFF | SV\_NOC\_SYS |
| 25 | 0x0F10\_0000 | 0xF\_FFFF | SV\_NOC\_VIP |
| 26 | 0x0F20\_0000 | 0xF\_FFFF | SV\_NOC\_DBG |
| 27 | 0x0F30\_0000 | 0xF\_FFFF | SV\_NOC\_MAC |
| 28 | 0x0F40\_0000 | 0xF\_FFFF | SV\_NOC\_SDC |

* 1. DMA Channel

Table 4.DMA peripheral

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IP #** | **IP Name** | **peripheral#0** | **peripheral#1** | **peripheral#2** | **peripheral#3** | **peripheral#4** | **peripheral#5** | **peripheral#6** | **peripheral#7** |
| **1** | **SPI#0-RX** | ● |  |  |  |  |  |  |  |
| **2** | **SPI#0-TX** |  | ● |  |  |  |  |  |  |
| **3** | **UART-RX** |  |  | ● |  |  |  |  |  |
| **4** | **UART-TX** |  |  |  | ● |  |  |  |  |
| **5** | **SPINOR-RX** |  |  |  |  | ● |  |  |  |
| **6** | **SPINOR-TX** |  |  |  |  |  | ● |  |  |
| **7** | **SPI#1-RX** |  |  |  |  |  |  | ● |  |
| **8** | **SPI#1-TX** |  |  |  |  |  |  |  | ● |

* 1. Interrupts Mapping

Table 5.Interrupts Mapping

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **IP Name** | **Interrupt Name** | **Link** | **ID** | **Trigger** | **Description** |
| - | - | - | 0~7 | edge | non-secure interrupts |
| - | - | - | 8~15 | edge | secure interrupts |
| - | - | - | 16~24 | - | - |
| CA7 | Virtual Maintenance Interrupt(PPI6) | - | 25 | level & low | Virtual Maintenance Interrupt(PPI6) |
| Hypervisor Timer event(PPI5) | nCNTHPIRQ | 26 | level & low | Hypervisor Timer event(PPI5) |
| Virtual Timer event(PPI4) | nCNTVIRQ | 27 | level & low | Virtual Timer event(PPI4) |
| Legacy nFIQ signal(PPI0) | nFIQ(tie 1) | 28 | level & low | Legacy nFIQ signal(PPI0) |
| Secure Physical Timer event(PPI1) | nCNTPSIRQ | 29 | level & low | Secure Physical Timer event(PPI1) |
| Non-secure Physical Timer event(PPI2) | nCNTPNSIRQ | 30 | level & low | Non-secure Physical Timer event(PPI2) |
| Legacy nIRQ signal(PPI3) | nIRQ(tie 1) | 31 | level & low | Legacy nIRQ signal(PPI3) |
| UART | INT\_UART\_uart\_int | IRQs[0] | 32 | level & high | From UART, when errors occur or FIFO trigger level reached or TX FIFO empty or RX FIFO full, this interrupt will assert. |
| PWM | INT\_PWM\_IO\_PWM\_PWM0\_INT | IRQs[1] | 33 | level & high | From PWM, when the control register value is equal to the value of capture PWM low or high , this interrupt assert |
| INT\_PWM\_IO\_PWM\_PWM1\_INT | IRQs[2] | 34 | level & high | From PWM, when the control register value is equal to the value of capture PWM low or high , this interrupt assert |
| INT\_PWM\_IO\_PWM\_PWM2\_INT | IRQs[3] | 35 | level & high | From PWM, when the control register value is equal to the value of capture PWM low or high , this interrupt assert |
| INT\_PWM\_IO\_PWM\_PWM3\_INT | IRQs[4] | 36 | level & high | From PWM, when the control register value is equal to the value of capture PWM low or high , this interrupt assert |
| I2C | INT\_I2C0\_i2c\_int | IRQs[5] | 37 | level & high | From I2C, when user choose master transmit or receive mode or slave receive or transmit mode, this interrupt assert. |
| INT\_I2C1\_i2c\_int | IRQs[6] | 38 | level & high | From I2C, when user choose master transmit or receive mode or slave receive or transmit mode, this interrupt assert. |
| TIMER | INT\_TIMER\_TMR0\_INT | IRQs[7] | 39 | level & high | From Timer, when counting down to zero, this interrupt assert. |
| INT\_TIMER\_TMR1\_INT | IRQs[8] | 40 | level & high | From Timer, when counting down to zero, this interrupt assert. |
| INT\_TIMER\_TMR2\_INT | IRQs[9] | 41 | level & high | From Timer, when counting down to zero, this interrupt assert. |
| SPI | INT\_SPI0\_spi\_int | IRQs[10] | 42 | level & high | From SPI0, when transmit or receive FIFO trigger leve is reached, or character length mismatch the setting,  An interrupt will be asserted to the CPU. |
| INT\_SPI1\_spi\_int | IRQs[11] | 43 | level & high | From SPI1, when transmit or receive FIFO trigger leve is reached, or character length mismatch the setting,  An interrupt will be asserted to the CPU. |
| GPIO | INT\_GPIO\_gpio\_int | IRQs[12] | 44 | level & high | From GPIO, either rising /falling edge or high/low level is detected on the corresponding port(A/B/C/D), An interrupt will be asserted . |
| SPINOR | INT\_SPINOR\_spi\_int | IRQs[13] | 45 | level & high | The signal means is that can be display tx/rx data transfer status (include character length and rx/tx fifo register status).If any situation is happen, the interrupt signal will be assert. |
| DMA | INT\_DMA\_irq\_OR | IRQs[14] | 46 | level & high | The DMA interrupt when it have interrupt event |
| GMAC | INT\_GMAC\_interrupt | IRQs[15] | 47 | level & high | Only one signal from GMAC controller. Some normal/abnormal interrupts such as “Transmit interrupt” and “Receive interrupt” assert this interrupt. Three registers, CSR5, CSR7 and CSR11 contain the interrupt status information. This signal is high-level active. |
| CRYPTO | INT\_CRYPTO\_O\_pka\_irq | IRQs[16] | 48 | level & high | From Crypto engine, when public key compute completely and output, this interrupt is assert. |
| INT\_CRYPTO\_O\_rng\_irq | IRQs[17] | 49 | level & high | From Crypto engine, when user choose random mode and the random generator in it generate a number，this interrupt assert. If user don’t choose this mode, this interrupt is invalid. |
| INT\_CRYPTO\_O\_spacc\_irq[0] | IRQs[18] | 50 | level & high | From Crypto engine, when three command fifos in spacc module are all full, this interrupt assert. |
| INT\_CRYPTO\_O\_irq | IRQs[19] | 51 | level & high | From Crypto engine, the summary of three interrupt include PKA, RNG, SPAcc .When one of them is assert ,this interrupt will assert. |
| NOC | INT\_NOC\_ObsErr1\_mainFault\_0 | IRQs[20] | 52 | level | Asserted when NOC got protocol error on CPU-NIU |
| INT\_NOC\_ObsErr0\_mainFault\_0 | IRQs[21] | 53 | level | Asserted when NOC got protocol error on DMA-NIU |
| ISRAM | INT\_ISRAM\_smc\_int | IRQs[22] | 54 | level & high | From ISRAM combined interrupt |
| INT\_ISRAM\_smc\_int0 | IRQs[23] | 55 | level & high | From ISRAM, individual memory interface 0 interrupt output |
| INT\_ISRAM\_smc\_int1 | IRQs[24] | 56 | level & high | From ISRAM, individual memory interface 1 interrupt output |
| SDIO | INT\_SDIO\_interrupt | IRQs[25] | 57 | level & high | The signal means is that can be display card situation (include command, data, current, Buffer, DMA and wakeup status).If any situation is happen, the interrupt signal will be assert. |
| INT\_SDIO\_wakeup | IRQs[26] | 58 | level & high | The signal means is that can be display card in slot or not (include card interrupt, removed and inserted).If any situation is happen, the interrupt signal will be assert. |
| DDR | INT\_DDR\_controller\_int | IRQs[27] | 59 | level & high | From DDR MC, when MC encounter some status(Initial done / Access out of range / Low power mode … etc), this interrupt is assert, active high. |
| PMU | INT\_PMU\_PUPPDN\_N | IRQs[28] | 60 | level & low | Interrupt signal from the PMU，power up power down interrupt |
| IVA | INT\_IVA\_INT | IRQs[29] | 61 | level & high | IVA interrupt |
| Cortex-A7 | nAXIERRIRQ | IRQs[170] | 202 | level & low | Error indicator for AXI transactions with an error condition that might not map to an individual processor in the multiprocessor device. |
| nPMUIRQ | IRQs[171] | 203 | level & low | CPU Performance Monitoring Unit interrupt signals |
| cti\_irq\_n | IRQs[172] | 204 | level & low |  |

1. Cortex-A7
   1. Register Summary
      1. Cortex-A7 c0 register summary

Table 6.Cortex-A7 c0 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| MIDR | 0x410FC075 | Main ID Register |
| CTR | 0x84448003 | Cache Type Register |
| TCMTR | 0x00000000 | TCM Type Register |
| TLBTR | 0x00000000 | TLB Type Register |
| MPIDR | -a | Multiprocessor Affinity Register |
| REVIDR | 0x00000000 | Revision ID Register |
| MIDR | 0x410FC075 | Aliases of Main ID Register |
| ID\_PFR0 | 0x00001131 | Processor Feature Register 0 |
| ID\_PFR1 | 0x00011011 | Processor Feature Register 1 |
| ID\_DFR0 | 0x02010555 | Debug Feature Register 0 |
| ID\_AFR0 | 0x00000000 | Auxiliary Feature Register 0 |
| ID\_MMFR0 | 0x10101105 | Memory Model Feature Register 0 |
| ID\_MMFR1 | 0x40000000 | Memory Model Feature Register 1 |
| ID\_MMFR2 | 0x01240000 | Memory Model Feature Register 2 |
| ID\_MMFR3 | 0x02102211 | Memory Model Feature Register 3 |
| ID\_ISAR0 | 0x01101110 | Instruction Set Attribute Register 0 |
| ID\_ISAR1 | 0x13112111 | Instruction Set Attribute Register 1 |
| ID\_ISAR2 | 0x21232041 | Instruction Set Attribute Register 2 |
| ID\_ISAR3 | 0x11112131 | Instruction Set Attribute Register 3 |
| ID\_ISAR4 | 0x10011142 | Instruction Set Attribute Register 4 |
| ID\_ISAR5 | 0x00000000 | Instruction Set Attribute Register 5 |
| CCSIDR | UNK | Cache Size ID Register |
| CLIDR | -b | Cache Level ID Register |
| AIDR | 0x00000000 | Auxiliary ID Register |
| CSSELR | UNK | Cache Size Selection Register |
| VPIDR | -c | Virtualization Processor ID Register |
| VMPIDR | -d | Virtualization Multiprocessor ID Register |

Note:

a. The reset value depends on the primary input CLUSTER ID and the CPU ID value in the Multiprocessor Affinity Register.

b. The reset value depends on whether L2 cache is implemented.

c. The reset value is the value of the Main ID Register.

d. The reset value is the value of the Multiprocessor Affinity Register.

* + 1. Cortex-A7 c1 register summary

Table 7.Cortex-A7 c1 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| SCTLR | 0x00C50878a | System Control Register |
| ACTLR | 0x00000000 | Auxiliary Control Register |
| CPACR | 0x00000000b | Coprocessor Access Control Register |
| SCR | 0x00000000 | Secure Configuration Register |
| SDER | UNK | Secure Debug Enable Register |
| NSACR | 0x00000000c | Non-Secure Access Control Register |
| HSCTLR | UNK | Hyp System Control Register |
| HACTLR | UNK | Hyp Auxiliary Configuration Register |
| HCR | 0x00000000 | Hyp Configuration Register |
| HDCR | 0x00000006d | Hyp Debug Control Register |
| HCPTR | 0x000033FFe | Hyp Coprocessor Trap Register |
| HSTR | 0x00000000 | Hyp System Trap Register |
| HACR | UNK | Hyp Coprocessor Trap Register |

Note:

a. The reset value depends on primary inputs, CFGTE, CFGEND, and VINITHI. The value shown in Table XX assumes these signals are set to zero.

b. The reset value depends on the FPU and NEON configuration. If FPU and Advanced SIMD are implemented, the reset value is 0x00000000.If FPU is implemented but Advanced SIMD is not implemented, the reset value is 0xC0000000. If FPU and Advanced SIMD are not implemented, the reset value is Unknown.

c. The reset value depends on the FPU and NEON configuration. If FPU and Advanced SIMD are implemented, the reset value is 0x00000000.If FPU is implemented but Advanced SIMD is not implemented, the reset value is 0x00008000.If FPU and Advanced SIMD are not implemented, the reset value is 0x00000000.

d. The reset value for bit 7 is UNK.

e. The reset value depends on the FPU and NEON configuration. If FPU and Advanced SIMD are implemented, the reset value is 0x000033FF. If FPU is implemented but Advanced SIMD is not implemented, the reset value is 0x0000B3FF. If FPU and Advanced SIMD are not implemented, the reset value is 0x0000BFFF.

* + 1. Cortex-A7 c2 register summary

Table 8.Cortex-A7 c2 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| TTBR0 | UNK | Translation Table Base Register 0 |
| TTBR1 | UNK | Translation Table Base Register 1 |
| TTBCR | 0x00000000a | Translation Table Base Control Register |
| HTCR | UNK | Hyp Translation Control Register |
| VTCR | UNK | Virtualization Translation Control Register |

Note:

The reset value is 0x00000000for the Secure copy of the register. The reset value for the EAE bit of the Non-secure copy of the register is 0x0.You must program the Non-secure copy of the register with the required initial value, as part of the processor boot sequence.

* + 1. Cortex-A7 c3 register summary

Table 9.Cortex-A7 c3 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| DACR | UNK | Domain Access Control Register |

* + 1. Cortex-A7 c5 register summary

Table 10.Cortex-A7 c5 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| DFSR | UNK | Data Fault Status Register |
| IFSR | UNK | Instruction Fault Status Register |
| ADFSR | UNK | Auxiliary Data Fault Status Register |
| AIFSR | 0x00000000 | Auxiliary Instruction Fault Status Register |
| HADFSR | UNK | Hyp Auxiliary Data Fault Status Syndrome Register |
| HAIFSR | UNK | Hyp Auxiliary Instruction Fault Status Syndrome Register |
| HSR | UNK | Hyp Syndrome Register |

* + 1. Cortex-A7 c6 register summary

Table 11.Cortex-A7 c6 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| DFAR | UNK | Data Fault Address Register |
| IFAR | UNK | Instruction Fault Address Register |
| HDFAR | UNK | Hyp Data Fault Address Register |
| HIFAR | UNK | Hyp Instruction Fault Address Register |
| HPFAR | UNK | Hyp IPA Fault Address Register |

* + 1. Cortex-A7 c7 register summary

Table 12.Cortex-A7 c7 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| NOP | UNK | No Operation, see the ARM Architecture Reference Manual |
| ICIALLUIS | UNK | Instruction cache invalidate all to PoUa Inner Shareable |
| BPIALLIS | UNK | Branch predictor invalidate all Inner Shareable |
| PAR | UNK | Physical Address Register |
| ICIALLU | UNK | Instruction cache invalidate all to PoU |
| ICIMVAU | UNK | Instruction cache invalidate by MVA to PoU |
| CP15ISB | UNK | Instruction Synchronization Barrier operation |
| BPIALL | UNK | Branch predictor invalidate all |
| BPIMVA | UNK | Branch predictor invalidate by MVA |
| DCIMVACb | UNK | Data cache invalidate by MVA to PoCc |
| DCISWd | UNK | Data cache invalidate line by set/way |
| ATS1CPR | UNK | Stage 1 current state PL1 read |
| ATS1CPW | UNK | Stage 1 current state PL1 write |
| ATS1CUR | UNK | Stage 1 current state unprivileged (PL0) read |
| ATS1CUW | UNK | Stage 1 current state unprivileged (PL0) write |
| ATS12NSOPR | UNK | Stages 1 and 2 Non-secure PL1 read |
| ATS12NSOPW | UNK | Stages 1 and 2 Non-secure PL1 write |
| ATS12NSOUR | UNK | Stages 1 and 2 Non-secure unprivileged (PL0) read |
| ATS12NSOUW | UNK | Stages 1 and 2 Non-secure unprivileged (PL0) write |
| DCCMVAC | UNK | Data cache clean line by MVA to PoC |
| DCCSW | UNK | Data cache clean line by set/way |
| CP15DSB | UNK | Data Synchronization Barrier operation |
| CP15DMB | UNK | Data Memory Barrier operation |
| DCCMVAU | UNK | Clean data cache line by MVA to PoU |
| NOP | UNK | No Operation |
| DCCIMVAC | UNK | Data cache clean and invalidate line by MVA to PoC |
| DCCISW | UNK | Data cache clean and invalidate line by set/way |
| ATS1HR | UNK | Add translation stage 1 Hyp mode read |
| ATS1HW | UNK | Add translation stage 1 Hyp mode write |

Note:

a. PoU = Point of Unification. If BROADCASTINNER is LOW, the PoU is the L1 data cache. If BROADCASTINNER is HIGH then the PoU is outside of the processor and is dependent on the external memory system..

b. DCIMVAC is upgraded to DCCIMVAC for the individual processor that the DCIMVAC is executed on. Additionally, if the DCIMVAC is executed from a Non-secure state other than Hyp mode without second state write permissions then the DCIMVAC is upgraded to DCCIMVAC when broadcast to other processors or broadcast on the ACE interface.

c. PoC = Point of Coherence. The PoC is always outside of the processor and is dependent on the external memory system.

d. DCISW is upgraded to DCCISW when executed in a Non-secure PL1 mode if HCR.SWIO is set to 1

* + 1. Cortex-A7 c8 register summary

Table 13.Cortex-A7 c8 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| TLBIALLIS | UNK | Invalidate entire TLB Inner Shareable |
| TLBIMVAIS | UNK | Invalidate unified TLB entry by MVA Inner Shareable |
| TLBIASIDIS | UNK | Invalidate unified TLB by ASID match Inner Shareable |
| TLBIMVAAIS | UNK | Invalidate unified TLB by MVA all ASID Inner Shareable |
| ITLBIALL | UNK | Invalidate instruction TLB |
| ITLBIMVA | UNK | Invalidate instruction TLB entry by MVA |
| ITLBIASID | UNK | Invalidate instruction TLB by ASID match |
| DTLBIALL | UNK | Invalidate data TLB |
| DTLBIMVA | UNK | Invalidate data TLB entry by MVA |
| DTLBIASID | UNK | Invalidate data TLB by ASID match |
| TLBIALL | UNK | Invalidate unified TLB |
| TLBIMVA | UNK | Invalidate unified TLB entry by MVA |
| TLBIASID | UNK | Invalidate unified TLB by ASID match |
| TLBIMVAA | UNK | Invalidate unified TLB by MVA all ASID |
| TLBIALLHIS | UNK | Invalidate entire Hyp Unified TLB Inner Shareable |
| TLBIMVAHIS | UNK | Invalidate Unified Hyp TLB entry by MVA Inner Shareable |
| TLBIALLNSNHIS | UNK | Invalidate entire NS Non-Hyp Unified TLB Inner Shareable |
| TLBIALLH | UNK | Invalidate entire Hyp Unified TLB |
| TLBIMVAH | UNK | Invalidate Unified Hyp TLB entry by MVA |
| TLBIALLNSNH | UNK | Invalidate entire NS Non-Hyp Unified TLB |

* + 1. Cortex-A7 c9 register summary

Table 14.Cortex-A7 c9 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| PMCR | 0x41072000 | Performance Monitor Control Register |
| PMNCNTENSET | UNK | Count Enable Set Register |
| PMNCNTENCLR | UNK | Count Enable Clear Register |
| PMOVSR | UNK | Overflow Flag Status Register |
| PMSWINC | UNK | Software Increment Register |
| PMSELR | UNK | Event Counter Selection Register |
| PMCEID0 | 0x3FFF0F3F | Common Event Identification Register 0 |
| PMCEID1 | 0x00000000 | Common Event Identification Register 1 |
| PMCCNTR | UNK | Cycle Count Register |
| PMXEVTYPER | UNK | Event Type Selection Register |
| PMXEVCNTR | UNK | Event Count Register |
| PMUSERENR | 0x00000000 | User Enable Register |
| PMINTENSET | UNK | Interrupt Enable Set Register |
| PMINTENCLR | UNK | Interrupt Enable Clear Register |
| PMOVSSET | UNK | Performance Monitor Overflow Flag Status Set Register |
| L2CTLR | 0x00000000a | L2 Control Register |
| L2ECTLR | 0x00000000 | L2 Extended Control Register |

* + 1. Cortex-A7 c10 register summary

Table 15.Cortex-A7 c10 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| PRRR | UNK | Primary Region Remap Register |
| MAIR0 | UNK | Memory Attribute Indirection Registers 0 |
| NMRR | UNK | Normal Memory Remap Register |
| MAIR1 | UNK | Memory Attribute Indirection Registers 1 |
| AMAIR0 | UNK | Auxiliary Memory Attribute Indirection Register 0 |
| AMAIR1 | UNK | Auxiliary Memory Attribute Indirection Register 1 |
| HMAIR0 | UNK | Hyp Memory Attribute Indirection Register 0 |
| HMAIR1 | UNK | Hyp Memory Attribute Indirection Register 1 |
| HAMAIR0 | UNK | Hyp Auxiliary Memory Attribute Indirection Register 0 |
| HAMAIR1 | UNK | Hyp Auxiliary Memory Attribute Indirection Register 1 |

* + 1. Cortex-A7 c12 register summary

Table 16.Cortex-A7 c12 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| VBAR | 0x00000000a | Vector Base Address Register |
| MVBAR | UNK | Monitor Vector Base Address Register |
| ISR | UNK | Interrupt Status Register |
| HVBAR | UNK | Hyp Vector Base Address Register |

* + 1. Cortex-A7 c13 register summary

Table 17.Cortex-A7 c13 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| FCSEIDR | 0x00000000 | FCSE Process ID Register |
| CONTEXTIDR | UNK | FCSE Process ID Register |
| TPIDRURW | UNK | User Read/Write Thread ID Register |
| TPIDRURO | UNK | User Read Only Thread ID Register |
| TPIDRPRW | UNK | Privileged Only Thread ID Register |
| HTPIDR | UNK | Hyp Software Thread ID Register |

* + 1. Cortex-A7 c14 register summary

Table 18.Cortex-A7 c14 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| CNTFRQ | UNK | Counter Frequency Register |
| CNTPCT | UNK | Counter Physical Count Register |
| CNTKCTL | -a | Counter Non-secure PL1 Control Register |
| CNTP\_TVAL | UNK | Counter PL1 Physical Timer Value Register |
| CNTP\_CTL | -b | Counter PL1 Physical Timer Control Register |
| CNTV\_TVAL | UNK | Counter PL1 Virtual Timer Value Register |
| CNTV\_CTL | -b | Counter PL1 Virtual Timer Control Register |
| CNTVCT | UNK | Counter Virtual Count Register |
| CNTP\_CVAL | UNK | Counter PL1 Physical Compare Value Register |
| CNTV\_CVAL | UNK | Counter PL1 Virtual Compare Value Register |
| CNTVOFF | UNK | Counter Virtual Offset Register |
| CNTHCTL | -c | Counter Non-secure PL2 Control Register |
| CNTHP\_TVAL | UNK | Counter Non-securePL2 Physical Timer Value Register |
| CNTHP\_CTL | -b | Counter Non-secure PL2 Physical Timer Control Register |
| CNTHP\_CVAL | UNK | Counter Non-secure PL2 Physical Compare Value Register |

Note:

a. The reset value for bits 2:0 is 0b000.

b. The reset value for bit 0 is 0.

c. The reset value for bit 2 is 0 and for bits 1:0 is 0b11.

* + 1. Cortex-A7 c15 register summary

Table 19.Cortex-A7 c15 register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| CDBGDR0 | UNK | Data Register 0 |
| CDBGDR1 | UNK | Data Register 1 |
| CDBGDR2 | UNK | Data Register 2 |
| CDBGDCT | UNK | Data Cache Tag Read Operation Register |
| CDBGICT | UNK | Instruction Cache Tag Read Operation Register |
| CDBGDCD | UNK | Data Cache Data Read Operation Register |
| CDBGICD | UNK | Instruction Cache Data Read Operation Register |
| CDBGTD | UNK | TLB Data Read Operation Register |
| CBAR | -b | Configuration Base Address Register |

Note:

a. See direct access to internal memory for information on how these registers are used.

b. The reset value depends on the primary input, PERIPHBASE39:15

* + 1. 64-bit CP15 system control register summary

Table 20.64-bit CP15 system control register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Reset Value** | **Description** |
| TTBR0 | UNK | Translation Table Base Register 0 |
| TTBR1 | UNK | Translation Table Base Register 1 |
| HTTBR | UNK | Hyp Translation Table Base Register |
| VTTBR | UNK | Virtualization Translation Table Base Register |
| PAR | UNK | Physical Address Register |
| CNTPCT | UNK | Counter Physical Count Register |
| CNTVCT | UNK | Counter Virtual Count Register |
| CNTP\_CVAL | UNK | Counter PL1 Physical Compare Value Register |
| CNTV\_CVAL | UNK | Counter PL1 Virtual Compare Value Register |
| CNTVOFF | UNK | Counter Virtual Offset Register |
| CNTHP\_CVAL | UNK | Counter Non-secure PL2 Physical Compare Value Register |

Note: a. The reset value for bits 55:48 is 0b00000000

* + 1. GIC memory map

Table .CP15 GIC memory map

|  |  |
| --- | --- |
| **Address range** | **Component a** |
| 0x0000- 0x0FFF | Reserved |
| 0x1000- 0x1FFF | Distributor |
| 0x2000- 0x3FFF | CPU interface |
| 0x4000- 0x4FFF | Virtual interface control, common base address |
| 0x5000- 0x5FFF | Virtual interface control, processor-specific base address |
| 0x6000- 0x7FFF | Virtual CPU interface |

Note:

a.The GIC registers are memory-mapped, and the base address is specified by PERIPHBASE 39:15. This input must be tied to a constant value. The PERIPHBASE value is sampled during reset into the Configuration Base Address (CBAR)

* + 1. Distributor register summary

Table .Distributor register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| GICD\_CTLR | 0x000 | 0x00000000a | Distributor Control Register |
| GICD\_TYPER | 0x004 | IMPLEMENTATION DEFINED | Interrupt Controller Type Register |
| GICD\_IIDR | 0x008 | 0x0100143B | Distributor Implementer Identification Register |
| GICD\_IGROUPRn | 0x080-0x0BC | 0x00000000 | Interrupt Group Registers b |
| GICD\_ISENABLERn | 0x100 | 0x0000FFFFd | Interrupt Set-Enable Registers |
| 0x104-0x13C | 0x00000000 |
| GICD\_ICENABLERn | 0x180 | 0x0000FFFFd | Interrupt Clear-Enable Registers |
| 0x184-0x1BC | 0x00000000 |
| GICD\_ISPENDRn | 0x200-0x23C | 0x00000000 | Interrupt Set-Pending Registers |
| GICD\_ICPENDRn | 0x280-0x2BC | 0x00000000 | Interrupt Clear-Pending Registers |
| GICD\_ISACTIVERn | 0x300-0x33C | 0x00000000 | Interrupt Set-Active Registers |
| GICD\_ICACTIVERn | 0x380-0x3BC | 0x00000000 | Interrupt Clear-Active Registers |
| GICD\_IPRIORITYRne | 0x400-0x5FC | 0x00000000 | Interrupt Priority Registers |
| GICD\_ITARGETSRn | 0x800-0x81C | -f | Interrupt Processor Targets Registers |
| 0x820-0x9FC | 0x00000000 |
| GICD\_ICFGRn | 0xC00 | 0xAAAAAAAAg | Interrupt configuration registers |
| 0xC04 | 0x55540000g |
| 0xC08-0xC7C | 0x55555555g |
| GICD\_PPISR | 0xD00 | 0x00000000 | Private Peripheral Interrupt Status Register |
| GICD\_SPISRn | 0xD04-0xD3C | 0x00000000 | Shared Peripheral Interrupt Status Registers |
| GICD\_SGIR | 0xF00 | - | Software Generated Interrupt Register |
| GICD\_CPENDSGIRn | 0xF10-0xF1C | 0x00000000 | SGI Clear-Pending Registers |
| GICD\_SPENDSGIRn | 0xF20-0xF2C | 0x00000000 | SGI Set-Pending Registers |
| GICD\_PIDR4 | 0xFD0 | 0x00000004 | Peripheral ID4 Register |
| GICD\_PIDR5 | 0xFD4 | 0x00000000 | Peripheral ID5 Register |
| GICD\_PIDR6 | 0xFD8 | 0x00000000 | Peripheral ID6 Register |
| GICD\_PIDR7 | 0xFDC | 0x00000000 | Peripheral ID7 Register |
| GICD\_PIDR0 | 0xFE0 | 0x00000090 | Peripheral ID0 Register |
| GICD\_PIDR1 | 0xFE4 | 0x000000B4 | Peripheral ID1 Register |
| GICD\_PIDR2 | 0xFE8 | 0x0000002B | Peripheral ID2 Register |
| GICD\_PIDR3 | 0xFEC | 0x00000000 | Peripheral ID3 Register |
| GICD\_CIDR0 | 0xFF0 | 0x0000000D | Component ID0 Register |
| GICD\_CIDR1 | 0xFF4 | 0x000000F0 | Component ID1 Register |
| GICD\_CIDR2 | 0xFF8 | 0x00000005 | Component ID2 Register |
| GICD\_CIDR3 | 0xFFC | 0x000000B1 | Component ID3 Register |

Note:

a. You cannot modify the EnableGrp0 bit if CFGSDISABLEis asserted.

b. This register is only accessible from a secure access.

c. Writes to bits corresponding to the SGIs are ignored.

d. The reset value for the register that contains the SGI and PPI interrupts is 0x0000FFFFbecause SGIs are always enabled. However, SGIs are Group 0 on reset, so the reset value for Non-secure reads is 0x00000000.

e. Writing to the GICD\_IPRIORITYR does not affect the priority ofan active interrupt.

f. The register that contains the SGI and PPI interrupts is read-only and the value is implementation defined. For Cortex-A7 configurations with only one processor, these registers are RAZ/WI.

g. The reset value for the register that contains the SGI interrupts is 0xAAAAAAAA. The reset value for the register that contains the PPI interrupts is 0x55540000. The reset value for the registers that contain the SPI interrupts is 0x55555555.

* + 1. CPU Interface register summary

Table .CPU Interface register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| GICC\_CTLR | 0x000 | 0x00000000 | CPU Interface Control Register |
| GICC\_PMRn | 0x004 | 0x00000000 | Interrupt Priority Mask Register |
| GICC\_BPR | 0x008 | 0x00000002(S)a  0x00000003(NS)b | Binary Point Register |
| GICC\_IAR | 0x000C | 0x000003FF | Interrupt Acknowledge Register |
| GICC\_EOIR | 0x0010 | -- | End Of Interrupt Register |
| GICC\_RPR | 0x0014 | 0x000000FF | Running Priority Register |
| GICC\_HPPIR | 0x0018 | 0x000003FF | Highest Priority Pending Interrupt Register |
| GICC\_ABPR | 0x001C | 0x00000003 | Aliased Binary Point Register |
| GICC\_AIAR | 0x0020 | 0x000003FF | Aliased Interrupt Acknowledge Register |
| GICC\_AEOIR | 0x0024 | -- | Aliased End of Interrupt Register |
| GICC\_AHPPIR | 0x0028 | 0x000003FF | Aliased Highest Priority Pending Interrupt Register |
| GICC\_APR0 | 0x00D0-0x00DC | 0x00000000 | Active Priority Register |
| GICC\_NSAPR0 | 0x00E0 | 0x00000000 | Non-secure Active Priority Register |
| GICC\_IIDR | 0x00FC | 0x0102143B | CPU Interface Identification Register |
| GICC\_DIR | 0x1000 | -- | Deactivate Interrupt Register |

Note:

a. (S) = Secure

b. (NS) = Non-secure

c. This register is only accessible from a Secure access.

* + 1. Virtual interface control register summary

Table .Virtual interface control register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| GICH\_HCR | 0x000 | 0x00000000 | Hypervisor Control Register |
| GICH\_VTR | 0x004 | 0x90000003 | VGIC Type Register |
| GICH\_VMCR | 0x008 | 0x004C0000 | Virtual Machine Control Register |
| GICH\_MISR | 0x0010 | 0x00000000 | Maintenance Interrupt Status Register |
| GICH\_EISR0 | 0x0020 | 0x00000000 | End of Interrupt Status Register |
| GICH\_ELSR0 | 0x0030 | 0x0000000F | Empty List register Status Register |
| GICH\_APR0 | 0x00F0 | 0x00000000 | Active Priorities Register |
| GICH\_LR0 | 0x0100 | 0x00000000 | List Register 0 |
| GICH\_LR1 | 0x0104 | 0x00000000 | List Register 1 |
| GICH\_LR2 | 0x0108 | 0x00000000 | List Register 2 |
| GICH\_LR3 | 0x010C | 0x00000000 | List Register 3 |

* + 1. Virtual CPU interface register summary

Table .Virtual CPU interface register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| GICV\_CTLR | 0x000 | 0x00000000 | VM Control Register |
| GICV\_PMR | 0x004 | 0x00000000 | VM Priority Mask Register |
| GICV\_BPR | 0x008 | 0x00000002 | VM Binary Point Register |
| GICV\_IAR | 0x00C | 0x000003FF | VM Interrupt Acknowledge Register |
| GICV\_EOIR | 0x0010 | -- | VM End Of Interrupt Register |
| GICV\_RPR | 0x0014 | 0x000000FF | VM Running Priority Register |
| GICV\_HPPIR | 0x0018 | 0x000003FF | VM Highest Priority Pending Interrupt Register |
| GICV\_ABPR | 0x001C | 0x00000003 | VM Aliased Binary Point Register |
| GICV\_AIAR | 0x0020 | 0x000003FF | VM Aliased Interrupt Acknowledge Register |
| GICV\_AEOIR | 0x0024 | -- | VM Aliased End of Interrupt Register |
| GICV\_AHPPIR | 0x0028 | 0x000003FF | VM Aliased Highest Priority Pending Interrupt Register |
| GICV\_APR0 | 0x00D0 | 0x00000000 | VM Active Priority Register |
| GICV\_IIDR | 0x00FC | 0x0102143B | VM CPU Interface Identification Register |
| GICV\_DIR | 0x1000 | -- | VM Deactivate Interrupt Register |

* + 1. Address mapping for debug trace components

Table .Address mapping for debug trace components

|  |  |
| --- | --- |
| **Address range** | **Component a** |
| 0x00000- 0x00FFF | ROM table |
| 0x01000- 0x0FFFF | Reserved |
| 0x10000- 0x10FFF | CPU 0 Debug |
| 0x11000- 0x11FFF | CPU 0 PMU |
| 0x12000- 0x12FFF | CPU 1 Debug |
| 0x13000- 0x13FFF | CPU 1 PMU |
| 0x14000- 0x14FFF | CPU 2 Debug |
| 0x15000- 0x15FFF | CPU 2 PMU |
| 0x16000- 0x16FFF | CPU 3 Debug |
| 0x17000- 0x17FFF | CPU 3 PMU |
| 0x18000- 0x18FFF | CPU 0 CTI |
| 0x19000- 0x19FFF | CPU 1 CTI |
| 0x1A000- 0x1AFFF | CPU 2 CTI |
| 0x1B000- 0x1BFFF | CPU 3 CTI |
| 0x1C000- 0x1CFFF | CPU 0 Trace |
| 0x1D000- 0x1DFFF | CPU 1 Trace |
| 0x1E000- 0x1EFFF | CPU 2 Trace |
| 0x1F000- 0x1FFFF | CPU 3 Trace |

Note: a. Indicates the mapped component if present otherwise reserved.

* + 1. CP14 debug Register summary

Table .CP14 debug Register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Offset** | **Description** |
| DBGDIDR | 0x000 | Debug Identification Register |
| DBGDSCR internal view | 0x004 | Debug Status and Control Register |
| -- | 0x008-0x010 | Reserved |
| DBGDTRTX internal view | 0x014 | Target to Host Data Transfer Register |
| DBGDTRRX internal view | Host to Target Data Transfer Register |
| DBGWFAR | 0x018 | UNK/SBZP |
| DBGVCR | 0x01C | Vector Catch Register |
| -- | 0x020 | Reserved |
| DBGECR | 0x024 | Event Catch Register |
| -- | 0x028 | Not implemented |
| -- | 0x02C | Not implemented |
| -- | 0x030-0x07C | Reserved |
| DBGDTRRX external view | 0x080 | Host to Target Data Transfer Register |
| DBGITR | 0x084 | Instruction Transfer Register |
| DBGPCSR | Program Counter Sampling Register |
| DBGDSCR external view | 0x088 | Debug Status and Control Register |
| DBGDTRTX external view | 0x08C | Target to Host Data Transfer Register |
| DBGDRCR | 0x090 | Debug Run Control Register |
| DBGEACR | 0x094 | Debug ExternalAuxiliary Control Register |
| -- | 0x098-0x09C | Reserved |
| DBGPCSR | 0x0A0 | Program Counter Sampling Register |
| DBGCIDSR | 0x0A4 | Context ID Sampling Register |
| DBGVIDSR | 0x0A8 | Virtualization ID Sampling Register |
| -- | 0x0AC-0x0FC | Reserved |
| DBGBVRn | 0x100-0x114 | Breakpoint Value Registers |
| -- | 0x118-0x11C | Reserved |
| -- | 0x120-0x13C | Reserved |
| DBGBCRn | 0x140-0x154 | Breakpoint Control Registers |
| -- | 0x158-0x17C | Reserved |
| DBGWVRn | 0x180-0x18C | Watchpoint Value Registers |
| -- | 0x190-0x1BC | Reserved |
| DBGWCRn | 0x1C0-0x1CC | Watchpoint Control Registers |
| -- | 0x1D0-0x1FC | Reserved |
| -- | 0x200 | Reserved |
| -- | 0x204-0x24C | Reserved |
| DBGBXVRn | 0x250-0x254 | Breakpoint Extended Value Registers |
| -- | 0x258-0x2FC | Reserved |
| DBGOSLAR | 0x300 | OS Lock Access Register |
| DBGOSLSR | 0x304 | OS Lock Status Register |
| -- | 0x308 | Not implemented |
| -- | 0x30C | Reserved |
| DBGPRCR | 0x310 | Device Power-down and Reset Control Register |
| DBGPRSR | 0x314 | Device Power down and Reset Status Register |
| -- | 0x318-0x03C | Reserved |
| -- | 0x400-0x7FC | Reserved |
| -- | 0x800-0x8FC | Reserved |
| -- | 0x900-0xCFC | Reserved |
| Processor ID registers | 0xD00-0xDFC | Processor ID registers |
| -- | 0xE00-0xE7C | Reserved |
| -- | 0xE80-0xEF4 | Reserved |
| DBGITMISCOUT | 0xEF8 | Integration Miscellaneous Signals Register |
| DBGITMISCIN | 0xEFC | Integration Miscellaneous Signals Input Register |
| DBGITCTRL | 0xF00 | Integration Mode Control Register |
| -- | 0xF04-0xF9C | Reserved |
| DBGCLAIMSET | 0xFA0 | Claim Tag Set Register |
| DBGCLAIMCLR | 0xFA4 | Claim Tag Clear Register |
| -- | 0xFA8-0xFAC | Reserved |
| DBGLAR | 0xFB0 | Lock Access Register |
| DBGLSR | 0xFB4 | Lock Status Register |
| DBGAUTHSTATUS | 0xFB8 | Authentication Status Register |
| -- | 0xFBC | Reserved |
| DBGDEVID2 | 0xFC0 | UNK |
| DBGDEVID1 | 0xFC4 | Debug Device ID Register 1 |
| DBGDEVID | 0xFC8 | Debug Device ID Register |
| DBGDEVTYPE | 0xFCC | Device Type Register |
| DBGPID4 | 0xFD0 | Debug Peripheral Identification Registers |
| DBGPID5-7 | 0xFD4-0xFDC | Reserved |
| DBGPID0 | 0xFE0 | Debug Peripheral Identification Registers |
| DBGPID1 | 0xFE4 |
| DBGPID2 | 0xFE8 |
| DBGPID3 | 0xFEC |
| DBGCID0 | 0xFF0 | Debug Component Identification Registers |
| DBGCID1 | 0xFF4 |
| DBGCID2 | 0xFF8 |
| DBGCID3 | 0xFFC |
| DBGDSCR internal view | -- | Debug Status and Control Register |
| DBGDTRRX internal view | -- | Host to Target Data Transfer |
| DBGDTRTX internal view | Target to Host Transfer |
| DBGDRAR (MRC) | -- | Debug ROM Address Register |
| DBGDRAR (MRRC) | Debug ROM Address Register |
| DBGOSDLR | -- | OS Double Lock Register |
| DBGDSAR (MRC) | -- | Debug Self Address Offset Register |
| DBGDSAR (MRRC) | Debug Self Address Offset Register |

* + 1. PMU Register summary

Table .PMU Register summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Offset** | **Description** |
| PMXEVCNTR0 | 0x000 | Event Count Register, see the ARM Architecture Reference Manual |
| PMXEVCNTR1 | 0x004 |
| PMXEVCNTR2 | 0x008 |
| PMXEVCNTR3 | 0x00C |
| -- | 0x010-0x078 | Reserved |
| PMCCNTR | 0x07C | Cycle Count Register, see the ARM Architecture Reference Manual |
| -- | 0x080-0x3FC | Reserved |
| PMXEVTYPER0 | 0x400 | Event Type Selection Register, see the ARM Architecture Reference Manual |
| PMXEVTYPER1 | 0x404 |
| PMXEVTYPER2 | 0x408 |
| PMXEVTYPER3 | 0x40C |
| -- | 0x410-0x478 | Reserved |
| PMXEVTYPER31 | 0x47C | Performance Monitors Event Type Select Register 31,see the ARM Architecture Reference Manual |
| -- | 0x480-0xBFC | Reserved |
| PMCNTENSET | 0xC00 | Count Enable Set Register, see the ARM Architecture Reference Manual |
| -- | 0xC04-0xC1C | Reserved |
| PMCNTENCLR | 0xC20 | Count Enable Clear Register, see the ARM Architecture Reference Manual |
| -- | 0xC24-0xC3C | Reserved |
| PMINTENSET | 0xC40 | Interrupt Enable Set Register, see the ARM Architecture Reference Manual |
| -- | 0xC44-0xC5C | Reserved |
| PMINTENCLR | 0xC60 | Interrupt Enable Clear Register, see the ARM Architecture Reference Manual |
| -- | 0xC64-0xC7C | Reserved |
| PMOVSR | 0xC80 | Overflow Flag Status Register, see the ARM Architecture Reference Manual |
| -- | 0xC84-0xC9C | Reserved |
| PMSWINC | 0xCA0 | Software Increment Register, see the ARM Architecture Reference Manual |
| -- | 0xCA4-0xDFC | Reserved |
| PMCFGR | 0xE00 | Performance Monitor Configuration Register, see the ARM Architecture Reference Manual |
| PMCR | 0xE04 | Performance Monitor Control Register |
| PMUSERENR | 0xE08 | User Enable Register, see the ARM Architecture Reference Manual |
| -- | 0xE0C-0xE1C | Reserved |
| PMCEID0 | 0xE20 | Common Event Identification Register 0, see the ARM Architecture Reference Manual |
| PMCEID1 | 0xE24 | Common Event Identification Register 1, see the ARM Architecture Reference Manual |
| -- | 0xE28-0xFAC | Reserved |
| PMLAR | 0xFB0 | Lock Access Register, see the ARM Architecture Reference Manual |
| PMLSR | 0xFB4 | Lock Status Register, see the ARM Architecture Reference Manual |
| PMAUTHSTATUS | 0xFB8 | Authentication Status Register, see the ARM Architecture Reference Manual |
| -- | 0xFBC-0xFC8 | Reserved |
| PMDEVTYPE | 0xFCC | Device Type Register |
| PMPID4 | 0xFD0 | Performance Monitors Peripheral Identification Registers |
| PMPID5 | 0xFD4 |
| PMPID6 | 0xFD8 |
| PMPID7 | 0xFDC |
| PMPID0 | 0xFE0 |
| PMPID1 | 0xFE4 |
| PMPID2 | 0xFE8 |
| PMPID3 | 0xFEC |
| PMCID0 | 0xFF0 | Performance Monitors Component Identification Registers |
| PMCID1 | 0xFF4 |
| PMCID2 | 0xFF8 |
| PMCID3 | 0xFFC |

* 1. Register Descriptions

Main ID Register

Table .MIDR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RO | 0x41 | Indicates the implementer code:  0x41 ARM |
| 23:20 | RO | 0x0 | Indicates the variant number of the processor. This is the major revision number n in the *rn* part of the *rnpn* description of the product revision status:  0x0 Major revision numbers. |
| 19:16 | RO | 0xF | Indicates the architecture code:  0xF ARMv7 |
| 15:4 | RO | 0xC07 | Indicates the primary part number:  0xC07 Cortex-A7 MPCore part number. |
| 3:0 | RO | 0x5 | Indicates the revision number of the processor. This is the minor revision number n in the *pn* part of  the *rnpn* description of the product revision status:  0x5 Minor revision number |

Cache Type Register

Table .CTR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:29 | RO | 0x4 | Indicates the CTR format:  0x4 ARMv7 format. |
| 28 | -- | -- | Reserved, RAZ |
| 27:24 | RO | 0x4 | Cache Write-Back granule. Log2of the number of words of the maximum size of memory that can be overwritten as a result of the eviction of a cache entry that has had a memory location in it modified:  0x4 Cache Write-Back granule size is 16 words. |
| 23:20 | RO | 0x4 | Exclusives Reservation Granule. Log2of the number of words of the maximum size of the reservation granule that has been implemented for the Load-Exclusive and Store-Exclusive instructions:  0x4 Exclusive reservation granule size is 16 words. |
| 19:16 | RO | 0x4 | Log2of the number of words in the smallest cache line of all the data and unified caches that the processor controls:  0x4 Smallest data cache line size is 16 words |
| 15:14 | RO | 0x2 | L1 instruction cache policy. Indicates the indexing and tagging policy for the L1 instruction cache:  b10 Virtually Indexed Physically Tagged(VIPT). |
| 13:4 | RO | -- | Reserved, RAZ |
| 3:0 | RO | 0x3 | Log2of the number of words in the smallest cache line of all the instruction caches that the processor controls.  0x3 Smallest instruction cache line size is 8 words |

TLB Type Register

Table . TLBTR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | -- | -- | Reserved |
| 0 | RO | 0x0 | Indicates whether the implementation has a unified TLB:  0x0 Processor has a unified TLB. |

Multiprocessor Affinity Register

Table .MPIDR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 0x1 | Indicates that the processor implements the Multiprocessing Extensions register format:  0x1 ARMv7 multi-processor format |
| 30 | RO | 0x0 | Indicates a Uniprocessor system, as distinct from processor 0 in a multiprocessor system:  0x0 Processor is part of a multiprocessor system. |
| 29:25 | RO | -- | Reserved, RAZ. |
| 24 | RO | 0x0 | Indicates whether the lowest level of affinity consists of logical processors that are implemented using a multi-threading type approach.  0x0 Processors are not implemented using a multi-threading approach. |
| 23:21 | -- | -- | Reserved, RAZ. |
| 11:8 | RO | TBD | Indicates the value read in the CLUSTERID configuration pin. It identifies each Cortex-A7 MPCore processor in a system with more than one processor present. That is, there are other processors in the multiprocessor system that might not be Cortex-A7 MPCore processors. |
| 7:2 | -- | -- | Reserved, RAZ. |
| 1:0 | RO | TBD | Indicates the processor number in the Cortex-A7 MPCore processor. For:  • One processor, the CPU ID is 0x0.  • Two processors, the CPU IDs are 0x0and 0x1.  • Three processors, the CPU IDs are 0x0, 0x1, and 0x2.  • Four processors, the CPU IDs are 0x0, 0x1, 0x2, and 0x3. |

Revision ID Register

Table .REVIDR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0 | Implementation-specific revision information. The reset value is determined by the specific Cortex-A7 MPCore implementation |

Processor Feature Register 0

Table .ID\_PFR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | -- | -- | Reserved |
| 15:12 | RO | 0x1 | Indicates support for Thumb Execution Environment (ThumbEE) instruction set:  0x1 ThumbEE instruction set implemented |
| 11:8 | RO | 0x1 | Indicates support for Jazelle extension:  0x1 Processor supports trivial implementation of Jazelle extension |
| 7:4 | RO | 0x3 | Indicates support for Thumb instruction set:  0x3 Processor supports Thumb encoding after the introduction of Thumb-2 technology, and for all 16-bit and 32-bit Thumb basic instructions. |
| 3:0 | RO | 0x1 | Indicates support for ARM instruction set:  0x1 Processor supports ARM instruction set. |

Processor Feature Register 1

Table .ID\_PFR1 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:20 | -- | -- | Reserved |
| 19:16 | RO | 0x1 | Indicates support for Generic Timer:  0x1 Processor supports Generic Timer. |
| 15:12 | RO | 0x1 | Indicates support for Virtualization Extensions:  0x1 Processor supports Virtualization Extensions. |
| 11:8 | RO | 0x0 | Indicates support for microcontroller programmers model:  0x0 Processor does not support microcontroller programmers model. |
| 7:4 | RO | 0x1 | Indicates support for Security Extensions. This includes support for Monitor mode and the SMC instruction:  0x1 Processor supports Security Extensions. |
| 3:0 | RO | 0x0 | Indicates support for the standard programmers model for ARMv4 and later. Model must support User, FIQ,IRQ, Supervisor, Abort, Undefined and System modes:  0x1 Processor supports the standard programmers model for ARMv4 and later. |

Debug Feature Register 0

Table .ID\_DFR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | -- | -- | Reserved |
| 27:24 | RO | 0x2 | Indicates support for performance monitor model:  0x2 Processor supports Performance Monitor Unit version 2(PMUv2) architecture. |
| 23:20 | RO | 0x0 | Indicates support for memory-mapped debug model for M profile processors:  0x0 Processor does not support M profile Debug architecture |
| 19:16 | RO | 0x1 | Indicates support for memory-mapped trace model:  0x1 Processor supports ARM trace architecture, with memory-mapped access. |
| 15:12 | R | 0x0 | Indicates support for coprocessor-based trace model:  0x0 Processor does not support ARM trace architecture, with CP14 access |
| 11:8 | RO | 0x5 | Indicates support for memory-mapped debug model:  0x5 Processor supports v7.1 Debug architecture, with memory-mapped  access. |
| 7:4 | RO | 0x5 | Indicates support for coprocessor-based Secure debug model:  0x5 Processor supports v7.1 Debug architecture, withCP14 access |
| 3:0 | RO | 0x5 | Indicates support for coprocessor-based debug model:  0x5 Processor supports v7.1 Debug architecture, withCP14 access. |

Memory Model Feature Register 0

Table .ID\_MMFR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 0x1 | Indicates the innermost share ability domain implemented:  0x1 Processor implements hardware coherency support. |
| 27:24 | RO | 0x0 | Indicates support for Fast Context Switch Extension (FCSE):  0x0 Processor does not support FCSE. |
| 23:20 | RO | 0x1 | Indicates support for Auxiliary registers:  0x1 Processor supports the Auxiliary Control Register only. |
| 19:16 | RO | 0x0 | Indicates support for TCMs and associated DMAs:  0x0 Processor does not support TCM |
| 15:12 | RO | 0x1 | Indicates the number of share ability levels implemented:  0x1 Processor implements two levels of share ability |
| 11:8 | RO | 0x1 | Indicates the outermost share ability domain implemented:  0x1 Processor supports hardware coherency. |
| 7:4 | RO | 0x0 | Indicates support for a Protected Memory System Architecture(PMSA):  0x0 Processor does not support PMSA |
| 3:0 | RO | 0x5 | Indicates support for a Virtual Memory System Architecture (VMSA).  0x5 Processor supports:  • VMSAv7, with support for remapping and the Access flag.  • Privileged Execute Never (PXN) within the first level descriptors.  • 64-bit address translation descriptors. |

Memory Model Feature Register 1

Table .ID\_MMFR1 bit assignments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31:28 | RO | 0x4 | | Indicates branch predictor management requirements:  0x4 Branch predictor requires no flushing at any time. |
| 27:24 | RO | 0x0 | | Indicates the supported L1 data cache test and clean operations, for Harvard or unified cache implementation:  0x0 Not supported. |
| 23:20 | RO | 0x0 | | Indicates the supported entire L1 cache maintenance operations, for a unified cache implementation:  0x0 Not supported. |
| 19:16 | RO | 0x0 | | Indicates the supported entire L1 cache maintenance operations, for a Harvard cache implementation:  0x0 Not supported |
| 15:12 | RO | 0x0 | | Indicates the supported L1 cache line maintenance operations by set/way, for a unified cache implementation:  0x0 Not supported. |
| 11:8 | RO | 0x0 | | Indicates the supported L1 cache line maintenance operations by set/way, for a Harvard cache implementation:  0x0 Not supported. |
| 7:4 | RO | 0x0 | | Indicates the supported L1 cache line maintenance operations by MVA, for a unified cache implementation:  0x0 Not supported. |
| 3:0 | RO | 0x0 | | Indicates the supported L1 cache line maintenance operations by MVA, for a Harvard cache implementation:  0x0 Not supported |

Memory Model Feature Register 2

Table .ID\_MMFR2 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:28 | RO | 0x0 | Indicates support for Hardware Access flag:  0x0 Not supported. |
| 27:24 | RO | 0x1 | Indicates support for Wait For Interrupt (WFI) stalling:  0x1 Processor supports WFI stalling. |
| 23:20 | RO | 0x2 | Indicates the supported CP15 memory barrier operations.  0x2 Processor supports:  • Data Synchronization Barrier (DSB).  • Instruction Synchronization Barrier (ISB).  • Data Memory Barrier (DMB). |
| 19:16 | RO | 0x4 | Indicates the supported TLB maintenance operations, for a unified TLB implementation.  0x4 Processor supports:  • Invalidate all entries in the TLB.  • Invalidate TLB entry by MVA.  • Invalidate TLB entries by ASID match.  • Invalidate TLB entries by MVA All ASID.  • Invalidate unified Hyp TLB entry by MVA.  • Invalidate entire Non-secure Non-Hyp unified TLB.  • Invalidate entire Hyp unified TLB. |
| 15:12 | RO | 0x0 | Indicates the supported TLB maintenance operations, for a Harvard TLB implementation:  0x0 Not supported |
| 11:8 | RO | 0x0 | Indicates the supported L1 cache maintenance range operations, for a Harvard cache implementation:  0x0 Not supported. |
| 7:4 | RO | 0x0 | Indicates the supported L1 cache background prefetch operations, for a Harvard cache implementation:  0x0 Not supported. |
| 3:0 | RO | 0x0 | Indicates the supported L1 cache foreground prefetch operations, for a Harvard cache implementation:  0x0 Not supported. |

Memory Model Feature Register 3

Table .ID\_MMFR3 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:28 | RO | 0x0 | Indicates support for supersections:  0x0 Processor supports supersections. |
| 27:24 | RO | 0x2 | Indicates the size of physicalmemory supported by the processor caches:  0x2 Processor caches support 40-bit memory address. |
| 23:20 | RO | 0x1 | Indicates whether translation table updates require a clean to the point of unification:  0x1 Updates to the translation tables do not require a clean to the point of unification to ensure visibility by subsequent translation table walks. |
| 19:16 | -- | -- | Reserved |
| 15:12 | RO | 0x2 | Indicates whether cache, TLB and branch predictor operations are broadcast:  0x2 Cache, TLB and branch predictor operations affect structures according to shareability and defined behavior of instructions. |
| 11:8 | RO | 0x2 | Indicates the supported branch predictor maintenance operations.  0x2 Processor supports:  • Invalidate entire branch predictor array.  • Invalidate branch predictor by MVA. |
| 7:4 | RO | 0x1 | Indicates the supported cache maintenance operations by set/way.  0x1 Processor supports:  • Invalidate data cache by set/way.  • Clean data cache by set/way.  • Clean and invalidate data cache by set/way. |
| 3:0 | RO | 0x1 | Indicates the supported cache maintenance operations by MVA.  0x1 Processor supports:  • Invalidate data cache by MVA.  • Clean data cache by MVA.  • Clean and invalidate data cache by MVA.  • Invalidate instruction cache by MVA.  • Invalidate all instruction cache entries. |

Instruction Set Attribute Register 0

Table .ID\_ISAR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:28 | RO | -- | Reserved |
| 27:24 | RO | 0x1 | Indicates support for Divide instructions:  0x2 Processor supports:  • SDIV and UDIV in the Thumb instruction set.  • SDIV and UDIV in the ARM instruction set. |
| 23:20 | RO | 0x1 | Indicates the supported Debug instructions:  0x1 Processor supports BKPT instruction. |
| 19:16 | RO | 0x0 | Indicates the supported Coprocessor instructions:  0x0 None supported, except for separately attributed architectures including CP15, CP14, and Advanced SIMD and VFP. |
| 15:12 | RO | 0x1 | Indicates the supported combined Compare and Branch instructions in the Thumb instruction set:  0x1 Processor supports CBNZ and CBZ instructions. |
| 11:8 | RO | 0x1 | Indicates the supported bit field instructions:  0x1 Processor supports BFC, BFI, SBFX, and UBFX instructions |
| 7:4 | RO | 0x1 | Indicates the supported Bit Counting instructions:  0x1 Processor supports CLZ instruction. |
| 3:0 | RO | 0x0 | Indicates the supported Swap instructions in the ARM instruction set:  0x0 SWPand SWPBinstructions supported |

Note:

a. The WPinstruction only produces a read followed by a write that are not locked on the bus, if enabled in the SCTLR. See the description of ISAR4:SWP\_frac

Processor Feature Register 1

Table .ID\_PFR1 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:20 | -- | -- | Reserved |
| 19:16 | RO | 0x1 | Indicates support for Generic Timer:  0x1 Processor supports Generic Timer. |
| 15:12 | RO | 0x1 | Indicates support for Virtualization Extensions:  0x1 Processor supports Virtualization Extensions. |
| 11:8 | RO | 0x0 | Indicates support for microcontroller programmers model:  0x0 Processor does not support microcontroller programmers model. |
| 7:4 | RO | 0x1 | Indicates support for Security Extensions. This includes support for Monitor mode and the SMC instruction:  0x1 Processor supports Security Extensions. |
| 3:0 | RO | 0x0 | Indicates support for the standard programmers model for ARMv4 and later. Model must support User, FIQ,IRQ, Supervisor, Abort, Undefined and System modes:  0x1 Processor supports the standard programmers model for ARMv4 and later. |

Debug Feature Register 0

Table .ID\_DFR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:28 | -- | -- | Reserved |
| 27:24 | RO | 0x2 | Indicates support for performance monitor model:  0x2 Processor supports Performance Monitor Unit version 2(PMUv2) architecture. |
| 23:20 | RO | 0x0 | Indicates support for memory-mapped debug model for M profile processors:  0x0 Processor does not support M profile Debug architecture |
| 19:16 | RO | 0x1 | Indicates support for memory-mapped trace model:  0x1 Processor supports ARM trace architecture, with memory-mapped access. |
| 15:12 | RO | 0x0 | Indicates support for coprocessor-based trace model:  0x0 Processor does not support ARM trace architecture, with CP14 access |
| 11:8 | RO | 0x5 | Indicates support for memory-mapped debug model:  0x5 Processor supports v7.1 Debug architecture, with memory-mapped  access. |
| 7:4 | RO | 0x5 | Indicates support for coprocessor-based Secure debug model:  0x5 Processor supports v7.1 Debug architecture, withCP14 access |
| 3:0 | RO | 0x5 | Indicates support for coprocessor-based debug model:  0x5 Processor supports v7.1 Debug architecture, withCP14 access. |

Memory Model Feature Register 0

Table .ID\_MMFR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:28 | RO | 0x1 | Indicates the innermost share ability domain implemented:  0x1 Processor implements hardware coherency support. |
| 27:24 | RO | 0x0 | Indicates support for Fast Context Switch Extension (FCSE):  0x0 Processor does not support FCSE. |
| 23:20 | RO | 0x1 | Indicates support for Auxiliary registers:  0x1 Processor supports the Auxiliary Control Register only. |
| 19:16 | RO | 0x0 | Indicates support for TCMs and associated DMAs:  0x0 Processor does not support TCM |
| 15:12 | RO | 0x1 | Indicates the number of share ability levels implemented:  0x1 Processor implements two levels of share ability |
| 11:8 | RO | 0x1 | Indicates the outermost shareability domain implemented:  0x1 Processor supports hardware coherency. |
| 7:4 | RO | 0x0 | Indicates support for a Protected Memory System Architecture(PMSA):  0x0 Processor does not support PMSA |
| 3:0 | RO | 0x5 | Indicates support for a Virtual Memory System Architecture (VMSA).  0x5 Processor supports:  • VMSAv7, with support for remapping and the Access flag.  • Privileged Execute Never (PXN) within the first level descriptors.  • 64-bit address translation descriptors. |

Memory Model Feature Register 1

Table .ID\_MMFR1 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 0x4 | Indicates branch predictor management requirements:  0x4 Branch predictor requires no flushing at any time. |
| 27:24 | RO | 0x0 | Indicates the supported L1 data cache test and clean operations, for Harvard or unified cache implementation:  0x0 Not supported. |
| 23:20 | RO | 0x0 | Indicates the supported entire L1 cache maintenance operations, for a unified cache implementation:  0x0 Not supported. |
| 19:16 | RO | 0x0 | Indicates the supported entire L1 cache maintenance operations, for a Harvard cache implementation:  0x0 Not supported |
| 15:12 | RO | 0x0 | Indicates the supported L1 cache line maintenance operations by set/way, for a unified cache implementation:  0x0 Not supported. |
| 11:8 | RO | 0x0 | Indicates the supported L1 cache line maintenance operations by set/way, for a Harvard cache implementation:  0x0 Not supported. |
| 7:4 | RO | 0x0 | Indicates the supported L1 cache line maintenance operations by MVA, for a unified cache implementation:  0x0 Not supported. |
| 3:0 | RO | 0x0 | Indicates the supported L1 cache line maintenance operations by MVA, for a Harvard cache implementation:  0x0 Not supported |

Memory Model Feature Register 2

Table .ID\_MMFR2 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 0x0 | Indicates support for Hardware Access flag:  0x0 Not supported. |
| 27:24 | RO | 0x1 | Indicates support for Wait For Interrupt (WFI) stalling:  0x1 Processor supports WFI stalling. |
| 23:20 | RO | 0x2 | Indicates the supported CP15 memory barrier operations.  0x2 Processor supports:  • Data Synchronization Barrier (DSB).  • Instruction Synchronization Barrier (ISB).  • Data Memory Barrier (DMB). |
| 19:16 | RO | 0x4 | Indicates the supported TLB maintenance operations, for a unified TLB implementation.  0x4 Processor supports:  • Invalidate all entries in the TLB.  • Invalidate TLB entry by MVA.  • Invalidate TLB entries by ASID match.  • Invalidate TLB entries by MVA All ASID.  • Invalidate unified Hyp TLB entry by MVA.  • Invalidate entire Non-secure Non-Hyp unified TLB.  • Invalidate entire Hyp unified TLB. |
| 15:12 | RO | 0x0 | Indicates the supported TLB maintenance operations, for a Harvard TLB implementation:  0x0 Not supported |
| 11:8 | RO | 0x0 | Indicates the supported L1 cache maintenance range operations, for a Harvard cache implementation:  0x0 Not supported. |
| 7:4 | RO | 0x0 | Indicates the supported L1 cache background prefetch operations, for a Harvard cache implementation:  0x0 Not supported. |
| 3:0 | RO | 0x0 | Indicates the supported L1 cache foreground prefetch operations, for a Harvard cache implementation:  0x0 Not supported. |

Memory Model Feature Register 3

Table .ID\_MMFR3 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 0x0 | Indicates support for supersections:  0x0 Processor supports supersections. |
| 27:24 | RO | 0x2 | Indicates the size of physicalmemory supported by the processor caches:  0x2 Processor caches support 40-bit memory address. |
| 23:20 | RO | 0x1 | Indicates whether translation table updates require a clean to the point of unification:  0x1 Updates to the translation tables do not require a clean to the point of unification to ensure visibility by subsequent translation table walks. |
| 19:16 | -- | -- | Reserved |
| 15:12 | RO | 0x2 | Indicates whether cache, TLB and branch predictor operations are broadcast:  0x2 Cache, TLB and branch predictor operations affect structures according to shareability and defined behavior of instructions. |
| 11:8 | RO | 0x2 | Indicates the supported branch predictor maintenance operations.  0x2 Processor supports:  • Invalidate entire branch predictor array.  • Invalidate branch predictor by MVA. |
| 7:4 | RO | 0x1 | Indicates the supported cache maintenance operations by set/way.  0x1 Processor supports:  • Invalidate data cache by set/way.  • Clean data cache by set/way.  • Clean and invalidate data cache by set/way. |
| 3:0 | RO | 0x1 | Indicates the supported cache maintenance operations by MVA.  0x1 Processor supports:  • Invalidate data cache by MVA.  • Clean data cache by MVA.  • Clean and invalidate data cache by MVA.  • Invalidate instruction cache by MVA.  • Invalidate all instruction cache entries. |

Instruction Set Attribute Register 0

Table .ID\_ISAR0 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | -- | Reserved |
| 27:24 | RO | 0x1 | Indicates support for Divide instructions:  0x2 Processor supports:  • SDIV and UDIV in the Thumb instruction set.  • SDIV and UDIV in the ARM instruction set. |
| 23:20 | RO | 0x1 | Indicates the supported Debug instructions:  0x1 Processor supports BKPT instruction. |
| 19:16 | RO | 0x0 | Indicates the supported Coprocessor instructions:  0x0 None supported, except for separately attributed architectures including CP15, CP14, and Advanced SIMD and VFP. |
| 15:12 | RO | 0x1 | Indicates the supported combined Compare and Branch instructions in the Thumb instruction set:  0x1 Processor supports CBNZ and CBZ instructions. |
| 11:8 | RO | 0x1 | Indicates the supported bit field instructions:  0x1 Processor supports BFC, BFI, SBFX, and UBFX instructions |
| 7:4 | RO | 0x1 | Indicates the supported Bit Counting instructions:  0x1 Processor supports CLZ instruction. |
| 3:0 | RO | 0x0 | Indicates the supported Swap instructions in the ARM instruction set:  0x0 SWPand SWPBinstructions supported |

Note:

a. The SWPinstruction only produces a read followed by a write that are not locked on the bus, ifenabled in the SCTLR.See the description of ISAR4:SWP\_frac

Primary Region Remap Register

Table .PRRR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 24+na | RW | UNK | Outer Shareable property mapping for memory attributes n, if the region is mapped as Normal Shareable. N is the value of the TEX0, C and B bits, see Table 4-54 on page 4-56. The possible values of each NOS n bit are:  0 Memory region is Outer Shareable.  1 Memory region is Inner Shareable. |
| 23:20 | RW | UNK | Reserved, UNK/SBZP |
| 19 | RW | UNK | Mapping of S = 1 attribute for Normal memory. This bit gives the mapped Shareable attribute for a region of memory that:  • Is mapped as Normal memory.  • Has the S bit set to 1.  The possible values of the bit are:  0 Region is not Shareable.  1 Region is Shareable. |
| 18 | RW | UNK | Mapping of S = 0 attribute for Normal memory. This bit gives the mapped Shareable attribute for a region of memory that:  • Is mapped as Normal memory.  • Has the S bit set to 0.  The possible values of the bit are the same as those given for the NS1 bit, bit19. |
| 17 | RW | UNK | Mapping of S = 1 attribute for Device memory. This bit gives the mapped Shareable attribute for a region of memory that:  • Is mapped as Device memory.  • Has the S bit set to 1.  Note: This field has no significance in the processor. |
| 16 | RW | UNK | Mapping of S = 0 attribute for Device memory. This bit gives the mapped Shareable attribute for a region of memory that:  • Is mapped as Device memory.  • Has the S bit set to 0.  Note: This field has no significance in the processor |
| 2n+1:2na | RW | UNK | Primary TEX mapping for memory attributes n. n is the value of the TEX0, C and B bits. This field defines the mapped memory type for a region with attributes n. The possible values of the field are:  0b00 Strongly-ordered.  0b01 Device.  0b10 Normal Memory.  0b11 Reserved, effect is UNPREDICTABLE. |

Note:

a. Where n is 0-7

Memory Attribute Indirection Registers 0 and 1

Table .MAIR0 and MAIR1 bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 7:0a | RW | UNK | The memory attribute encoding for an AttrIndx2:0 entry in a Long descriptor format translation table entry, where:  • AttrIndx2 selects the appropriate MAIR:  — Setting AttrIndx2 to 0 selects MAIR0.  — Setting AttrIndx2 to 1 selects MAIR1.  • AttrIndx2:0 gives the value of min Attrm. |

Note:

a. where m is 0-7

Normal Memory Remap Register

Table .NMRR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 2n+17:2n+16a | RW | UNK | Outer Cacheable property mapping for memory attributes n, if the region is mapped as Normal memory by the PRRR. TRn entry. n is the value of the TEX0, C and B bits. The possible values of this field are:  0b00 Region is Non-cacheable.  0b01 Region is Write-Back, Write-Allocate.  0b10 Region is Write-Through, no Write-Allocate.  0b11 Region is Write-Back, no Write-Allocate. |
| 2n+1:2na | RW | UNK | Inner Cacheable property mapping for memory attributes n, if the region is mapped as Normal Memory by the PRRR. TRn entry. N is the value of the TEX0, C and B bits. The possible values of this field are the same as those given for the ORn field |

Note:

a. where m is 0-7

Auxiliary Control Register

Table .ACTLR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:29 | - | 0x0 | Reserved, RAZ/WI. |
| 28 | RW | 0x0 | Disable dual issue:  0 Enables dual issue, this is the reset value.  1 Disables dual issue. |
| 27:16 | - | 0x0 | Reserved, RAZ/WI |
| 15 | RW | 0x0 | Disable Distributed Virtual Memory (DVM) transactions:  0 Enables DVM, this is the reset value.  1 Disables DVM. |
| 14:13 | RW | 0x0 | L1 Data prefetch control. The value of this field determines the maximum number of outstanding data  prefetches permitted in the L1 memory system, not counting those generated by software load or PLD instructions:  0b00 Prefetch disabled.  0b01 1 outstanding pre-fetch permitted.  0b10 2 outstanding pre-fetches permitted.  0b11 3 outstanding pre-fetches permitted, this is the reset value. |
| 12 | RW | 0x0 | L1 Data Cache read-allocate mode :  0 Enables L1 data cache read-allocate mode, this is the reset value.  1 Disables L1 data cache read-allocate mode. |
| 11 | RW | 0x0 | L2 Data Cache read-allocate mode disable a:  0 Enables L2 data cache read-allocate mode, this is the reset value.  1 Disables L2 data cache read-allocate mode. |
| 10 | RW | 0x0 | Disable optimized data memory barrier behavior:  0 Enables optimized data memory barrier behavior, this is the reset value.  1 Disables optimized data memory barrier. |
| 9:7 | - | 0x0 | Reserved, RAZ/WI. |
| 6 | RW | 0x0 | Enables coherent requests to the processor:  0 Disables coherent requests to the processor. This is the reset value.  1 Enables coherent requests to the processor.  When coherent requests are disabled:  • Loads to cacheable memory are not cached by the processor.  • Load-Exclusive instructions take a precise abort if the memory attributes are:  — Inner Write-Back and Outer Shareable.  — Inner Write-Through and Outer Shareable.  — Outer Write-Back and Outer Shareable.  — Outer Write-Through and Outer Shareable.  — Inner Write-Back and Inner Shareable.  — Inner Write-Through and Inner Shareable.  — Outer Write-Back and Inner Shareable.  — Outer Write-Through and Inner Shareable.  Note: You must ensure this bit is set to 1 before the caches and MMU are enabled, or any cache and TLB Maintenance operations are performed. The only time this bit is set to 0 is during a processor power-down sequence |
| 5:0 | - | 0x0 | Reserved, RAZ/WI |

Note:

a. See Data Cache Unit for more information

Coprocessor Access Control Register

Table .CPACR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RW | -a | Disable Advanced SIMD Functionality:  0 All Advanced SIMD and VFP instructions execute normally.  1 All Advanced SIMD instructions executed take an Undefined instruction exception.  See the Cortex-A7 MPCore Floating-Point Unit Technical Reference Manual and Cortex-A7 MPCore NEON Media Processing Engine Technical Reference Manual for more information.  If FPU is implemented and Advanced SIMD is not implemented, this bit is RAO/WI.  If FPU and Advanced SIMD are not implemented, this bit is UNK/SBZP. |
| 30 | RW | -a | Disable use of registers D16-D31 of the VFP register file:  0 All instructions accessing D0-D31 execute normally.  1 Any VFP instruction that attempts to access any of registers D16-D31 is UNDEFINED.  See the Cortex-A7 MPCore Floating-Point Unit Technical Reference Manual and Cortex-A7 MPCore NEON Media Processing Engine Technical Reference Manual for more information.  If FPU is implemented and Advanced SIMD is implemented, ARM deprecates writing a value that is not zero to this bit.  If FPU is implemented and Advanced SIMD is not implemented, this bit is RAO/WI.  If FPU and Advanced SIMD are not implemented, this bit is UNK/SBZP. |
| 29:24 | -- | -a | Reserved, RAZ/WI |
| 23:22 | RW | -a | Defines the access rights for coprocessor 11:  0b00 Access denied.Attempted accesses generate an Undefined Instruction exception. This is the reset value.  0b01 Access at PL1 or higher only.Attempted accesses in User mode generate an Undefined Instruction exception. Any attempt to access the coprocessor from software executing at PL0 generates an Undefined Instruction exception.  0b10 Reserved.The effect of this value is UNPREDICTABLE.  0b11 Full access.  If FPU and Advanced SIMD are not implemented, this field is RAZ/WI. |
| 21:20 | RW | -a | Defines the access rights for coprocessor 10:  0b00 Access denied. Attempted accesses generate an Undefined Instruction exception. This is the reset value.  0b01 Access at PL1 or higher only. Any attempt to access the coprocessor from software executing at PL0 generates an Undefined Instruction exception.  0b10 Reserved. The effect of this value is UNPREDICTABLE.  0b11 Full access.  If FPU and Advanced SIMD are not implemented, this field is RAZ/WI. |
| 19:0 | RW | -- | Reserved, RAZ/WI |

Note:

a.The reset value depends on the FPU and NEON configuration. If FPU and Advanced SIMD are implemented, the reset value is 0x00000000. If FPU is implemented but Advanced SIMD is not implemented, the reset value is 0xC0000000. If FPU and Advanced SIMD are not implemented, the reset value is Unknown

Secure Configuration Register

Table .SCR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | -- | -- | Reserved, UNK/SBZP |
| 9 | RW | 0x0 | Secure Instruction Fetch bit:  0 Secure state instruction fetches from Non-secure memory are permitted, this is the reset value.  1 Secure state instruction fetches from Non-secure memory are not permitted. |
| 8 | RW | 0x0 | Hyp Call enable.This bit enables the use of HVC instruction from Non-secure PL1 modes:  0 The HVC instruction is UNDEFINED in Non-secure PL1 mode, this is the reset value.  1 The HVC instruction is enabled in Non-secure PL1 mode, and performs a Hyp Call |
| 7 | RW | 0x0 | Secure Monitor Call disable. This bit causes the SMC instruction to be UNDEFINED in Non-secure state:  0 The SMC instructions executes normally in Non-Secure state, and performs a Secure Monitor Call, this is the reset value.  1 The SMC instruction is UNDEFINED in Non-secure state. |
| 6 | RW | 0x0 | Not Early Termination. This bit disables early termination of data operations.  This bit is not implemented, UNK/SBZP. |
| 5 | RW | 0x0 | A bit writable. This bit controls whether the A bit in the CPSR can be modified in Non-secure state:  0 The CPSR.A bit can be modified only in Secure state, this is the reset value.  1 The CPSR.A bit can be modified in any security state. |
| 4 | RW | 0x0 | F bit writable. This bit controls whether the F bit in the CPSR can be modified in Non-secure state:  0 The CPSR.F bit can be modified only in Secure state, this is the reset value.  1 The CPSR.F bit can be modified in any security state. |
| 3 | RW | 0x0 | External Abort handler. This bit controls which mode takes external aborts:  0 External aborts taken in Abort mode, this is the reset value.  1 External abort taken in Monitor mode. |
| 2 | RW | 0x0 | FIQ handler. This bit controls which mode takes FIQ exceptions:  0 FIQs taken in FIQ mode, this is the reset value.  1 FIQs taken in Monitor mode. |
| 1 | RW | 0x0 | IRQ handler. This bit controls which mode takes IRQ exceptions:  0 IRQs taken in IRQ mode, this is the reset value.  1 IRQs taken in Monitor mode. |
| 0 | RW | 0x0 | Non Secure bit. Except when the processor is in Monitor mode, this bit determines the security state of the processor.  Secure, this is the reset value.  1 Non-secure.  Note: When the processor is in Monitor mode, it is always in Secure state, regardless of the value of the NS bit. The value of the NS bit also affects the accessibility of the Banked CP15 registers in Monitor mode. |

Non-Secure Access Control Register

Table .NSACR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:20 | -- | -- | Reserved, UNK/SBZP |
| 19 | RW | -- | Reserved, RAZ/WI. |
| 18 | RW | 0x0 | Determines if the SMP bit of the Auxiliary Control Register(ACTLR) is writable in Non-secure state:  0 A write to ACTLR in Non-secure state is write-ignored. This is the reset value.  1 A write to ACTLR in Non-secure state can modify the value of the SMP bit. Other bits in the ACTLR are write-ignored. |
| 17 | RW | 0x0 | NS\_L2ERR Determines if the L2 AXI asynchronous error bit of the L2 Extended Control Register (L2ECTLR), are writable in Non-secure state:  0 A write to L2ECTLR in Non-secure state is ignored. This is the reset value.  1 A write to L2ECTLR in Non-secure state can modify the value of the L2 AXI asynchronous error bit. Other bits in the L2ECTLR are write-ignored. |
| 16 | RW | 0x0 | Reserved, RAZ/WI |
| 15 | RW | 0x0 | Disable Non-secure Advanced SIMD functionality:  0 This bit has no effect on the ability to write CPACR.ASEDIS, this is the reset value.  1 When executing in Non-secure state, the CPACR.ASEDIS bit has a fixed value of 1 and writes to it are ignored.  If FPU is implemented and Advanced SIMD is not implemented, this bit is RAO/WI.  If FPU and Advanced SIMD are not implemented, this bit is UNK/SBZP. |
| 14 | RW | 0x0 | Disable the Non-secure use of D16-D31 of the VFP register file:  0 This bit has no effect on the ability to write CPACR. D32DIS. This is the reset value.  1 The CPACR.D32DIS bit when executing in Non-secure state has a fixed value of 1 and writes to it are ignored.  If FPU is implemented and Advanced SIMD is implemented, ARM deprecates writing a value that is not zero to this bit.  If FPU is implemented and Advanced SIMD is not implemented, this bit is RAO/WI.  If FPU and Advanced SIMD are not implemented, this bit is UNK/SBZP. |
| 13:12 | -- | -- | Reserved, RAZ/WI. |
| 11 | RW | 0x0 | Non-secure access to coprocessor 11 enable:  0 Secure access only.Any attempt to access coprocessor 11 in Non-secure state results in an Undefined Instruction exception. If the processor is in Non-secure state, the corresponding bits in the CPACR ignore writes and read as 0b00, access denied. This is the reset value.  1 Secure or Non-secure access.  If FPU and Advanced SIMD are not implemented, this bit is RAZ/WI. |
| 10 | RW | 0x0 | Non-secure access to coprocessor 10 enable:  0 Secure access only.Any attempt to access coprocessor 10 in Non-secure state results in an Undefined Instruction exception. If the processor is in Non-secure state, the corresponding bits in the CPACR ignore writes and read as 0b00, access denied. This is the reset value.  1 Secure or Non-secure access.  If FPU and Advanced SIMD are not implemented, this bit is RAZ/WI. |
| 9:0 | RW | -- | Reserved, RAZ/WI. |

Hyp System Control Register

Table .HSCTLR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | -- | -- | Reserved, RAZ/WI. |
| 30 | RW | UNK | Thumb Exception enable.This bit controls whether exceptions taken in Hyp mode are taken in ARM or Thumb state:  0 Exceptions taken in ARM state.  1 Exceptions taken in Thumb state. |
| 29:28 | -- | -- | Reserved, RAO/WI. |
| 27:26 | -- | -- | Reserved, RAZ/WI. |
| 25 | RW | UNK | Exception Endianness bit. The value of this bit defines the value of the CPSR.E bit on entry to an exception vector in Hyp mode. This value also indicates the endianness of the translation table data for translation table lookups, when executing in Hyp mode:  0 Little endian.  1 Big endian. |
| 24 | -- | -- | Reserved, RAZ/WI. |
| 23:22 | -- | -- | Reserved, RAO/WI |
| 21 | RW | UNK | Fast Interrupts configuration enable bit. This bit can be used to reduce interrupt latency by disabling implementation-defined performance features.  This bit is not implemented, RAZ/WI |
| 20 | -- | -- | Reserved, RAZ/WI |
| 19 | RW | UNK | Write permission implies Execute Never(XN):  0 Hyp translations that permit write are not forced to be XN.  1 Hyp translations that permit write are forced to be XN. |
| 18 | -- | -- | Reserved, RAO/WI |
| 17 | -- | -- | Reserved, RAZ/WI. |
| 16 | -- | -- | Reserved, RAO/WI |
| 15:13 | -- | -- | Reserved, RAZ/WI. |
| 12 | RW | UNK | Instruction cache enable bit for memory accesses made in Hyp mode:  0 Instruction caches disabled.  1 Instruction caches enabled. |
| 11 | -- | -- | Reserved, RAO/WI |
| 10:7 | -- | -- | Reserved, RAO/WI. |
| 6:3 | -- | -- | Reserved, RAO/WI. |
| 2 | RW | UNK | Data and unified cache enable bit for memory accesses made in Hyp mode:  0 Data and unified caches disabled.  1 Data and unified caches enabled. |
| 1 | RW | UNK | Alignment fault checking enable bit for memory accesses made in Hyp mode:  0 Alignment fault checking disabled.  1 Alignment fault checking enabled. |
| 0 | RW | UNK | MMU stage 1 address translation enable bit for memory accesses made in Hyp mode:  0 Address translation disabled.  1 Address translation enabled. |

Hyp Debug Control Register

Table .HDCR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | -- | -- | Reserved, UNK/SBZP. |
| 11 | RW | 0x0 | Trap Debug ROM Access:  0 Has no effect on Debug ROM accesses.  1 Trap valid Non-secure Debug ROM accesses to Hyp mode.  When this bit is set to 1, any valid Non-secure access to the DBGDRAR or DBGDSAR is trapped to Hyp mode.  If this bit is set to 0 when TDA or TDE is set to 1, behavior is UNPREDICTABLE. This bit resets to 0. |
| 10 | RW | 0x0 | Trap Debug OS-related register Access:  0 Has no effect on accesses to CP14 Debug registers.  1 Trap valid Non-secure accesses to CP14 OS-related Debug registers to Hyp mode.  When this bit is set to 1, any valid Non-secure CP14 access to the following OS-related Debug registers is trapped to Hyp mode:  • DBGOSLSR.  • DBGOSLAR.  • DBGOSDLR.  • DBGPRCR.  If this bit is set to 0 when TDA or TDE is set to 1, behavior is UNPREDICTABLE. This bit resets to 0 |
| 9 | RW | 0x0 | Trap Debug Access:  0 Has no effect on accesses to CP14 Debug registers.  1 Trap valid Non-secure accesses toCP14 Debug registers to Hyp mode.  If this bit is set to 1, then the TDRA and TDOSA bits must also be set to 1, otherwise the behavior is UNPREDICTABLE  If this bit is set to 0 when TDE is set to 1, behavior is UNPREDICTABLE. This bit resets to 0. |
| 8 | RW | 0x0 | Trap Debug Exceptions:  0 Has no effect on Debug exceptions.  1 Trap valid Non-secure Debug exceptions to Hyp mode.  When this bit is set to 1:  • Any Debug exception taken in Non-secure state is trapped to Hyp mode.  • The TDRA, TDOSA, and TDA bits must all be set to 1, otherwise behavior is UNPREDICTABLE. This bit resets to 0. |
| 7 | RW | UNK | Hypervisor Performance Monitor Enable:  0 Hyp mode performance monitor counters disabled.  1 Hyp mode performance monitor counters enabled.  When this bit is set to 1, access to the performance monitors that are reserved for use from Hyp mode is enabled. For more information, see the description of the HPMN field.  The reset value of this bit is UNKNOWN. |
| 6 | RW | 0x0 | Trap Performance Monitor accesses:  0 Has no effect on performance monitor accesses.  1 Trap valid Non-secure performance monitor accesses to Hyp mode.  When this bit is set to 1, any valid Non-secure access to the Performance Monitor registers is trapped to Hyp mode. This bit resets to 0. See the ARM Architecture Reference Manual for more information |
| 5 | RW | 0x0 | Trap Performance Monitor Control Register accesses:  0 Has no effect on PMCR accesses.  1 Trap valid Non-secure PMCR accesses to Hyp mode.  When this bit is set to 1, any valid Non-secure access to the PMCR is trapped to Hyp mode. This bit resets to 0. See the ARM Architecture Reference Manual for more information. |
| 4:0 | RW | 0x6 | Hyp Performance Monitor count. Specifies the number of performance monitor counters that are accessible from Non-secure PL1 modes.  In Non-secure state, HPMN divides the performance monitor counters.  For example, If PMnEVCNTR is performance monitor counter n then, in Non-secure state:  •If n is in the range 0 ≤ n< HPMN, the counter is accessible from PL1 and PL2, and from PL0 if unprivileged access to the counters is enabled.  •If n is in the range HPMN ≤ n<PMCR.N, the counter is accessible only from PL2. The HPME bit enables access to the counters in this range.  Behavior of the Performance Monitors counters is UNPREDICTABLE if this field is set to a value greater than PMCR.N.  This field resets to 0x4, which is the value of PMCR.N. |

Note:

a. The permitted values of the HDCR.{TDRA, TDOSA, TDA, TDE} bits are 0b0000, 0b0100, 0b1000, 0b1100, 0b1110, and 0b1111. If these bits are set to any other values, behavior is UNPREDICTABLE.

Hyp Coprocessor Trap Register

Table .HCPTR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RW | 0x0 | Trap Coprocessor Access Control Register accesses:  0 Has no effect on CPACR accesses.  1 Trap valid Non-secure PL1 CPACR accesses to Hyp mode.  When this bit is set to 1, any valid Non-secure PL1 or PL0 access to the CPACR is trapped to Hyp mode. See the ARM Architecture Reference Manual for more information. |
| 30:16 | RW | 0x0 | Reserved, UNK/SBZP |
| 15 | RW | -a | Trap Advanced SIMD Extension:  0 If the NSACR settings permit Non-secure use of the Advanced SIMD functionality then Hyp mode can access that functionality, regardless of any settings in the CPACR  Note: This bit value has no effect on possible use of the Advanced SIMD functionality from Non-secure PL1 and PL0 modes.  1 Trap valid Non-secure accesses to Advanced SIMD functionality to Hyp mode.  When this bit is set to 1, any otherwise-valid access to Advanced SIMD functionality from:  • A Non-secure PL1 or PL0 mode is trapped to Hyp mode.  • Hyp mode generates an Undefined Instruction exception, taken in Hyp mode.  If FPU is implemented and Advanced SIMD is not implemented, this bit is RAO/WI.  If FPU and Advanced SIMD are not implemented, this bit is RAO/WI.  If NSACR.NSASEDIS is set to 1, then on Non-secure accesses to the HCPTR, the TASE bit behaves as RAO/WI. |
| 14 | RW | -a | Reserved, RAZ/WI. |
| 13:12 | RW | -a | Reserved, RAO/WI. |
| 11 | RW | -a | Trap coprocessor 11:  0 If NSACR.CP11 is set to 1, then Hyp mode can access CP11, regardless of the value of CPACR.CP11.  Note: This bit value has no effect on possible use of CP11 from Non-secure PL1 and PL0 modes.  1 Trap valid Non-secure accesses to CP11 to Hyp mode.  When TCP11 is set to 1, any otherwise-valid access to CP11 from:  • A Non-secure PL1 or PL0 mode is trapped to Hyp mode.  • Hyp mode generates an Undefined Instruction exception, taken in Hyp mode.  If VFP and Advanced SIMD are not implemented, this bit is RAO/WI. See the ARM Architecture Reference Manual for more information. |
| 10 | RW | -a | Trap coprocessor 10:  0 If NSACR.CP10 is set to 1, then Hyp mode can access CP10, regardless of the value of CPACR.CP10.  Note: This bit value has no effect on possible use of CP10 from Non-secure PL1 and PL0 modes.  1 Trap valid Non-secure accesses to CP10 to Hyp mode. When TCP10 is set to 1, any otherwise-valid access to CP10 from:  • A Non-secure PL1 or PL0 mode is trapped to Hyp mode.  • Hyp mode generates an Undefined Instruction exception, taken in Hyp mode.  If VFP and Advanced SIMD are not implemented, this bit is RAO/WI. See the ARM Architecture Reference Manual for more information. |
| 9:0 | RW | -a | Reserved, RAO/WI. |

Note:

a. The reset value depends on the FPU and NEON configuration. If FPU and Advanced SIMD are implemented, the reset value is 0x000033FF. If FPU is implemented but Advanced SIMD is not implemented, the reset value is 0x0000B3FF. If FPU and Advanced SIMD are not implemented, the reset value is 0x0000BFFF

Data Fault Status Register

Table .DFSR bit assignments for Short-descriptor translation table format

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset value** | **Description** |
| 31:14 | -- | -- | Reserved, UNK/SBZP. |
| 13 | RW | UNK | Cache maintenance fault. For synchronous faults, this bit indicates whether a cache maintenance operation generated the fault:  0 Abort not caused by a cache maintenance operation.  1 Abort caused by a cache maintenance operation. |
| 12 | RW | UNK | External abort type. This field indicates whether an AXI Decode or Slave error caused an abort:  0 External abort marked as DECERR.  1 External abort marked as SLVERR.  For aborts other than external aborts this bit always returns 0. |
| 11 | RW | UNK | Write not Read bit. This field indicates whether the abort was caused by a write or a read access:  0 Abort caused by a read access.  1 Abort caused by a write access.  For faults on CP15 cache maintenance operations, including the VA to PA translation operations, this bit always returns a value of 1. |
| 10 | RW | UNK | Part of the Fault Status field. See bits 3:0 in this table. |
| 9 | RW | UNK | RAZ. |
| 8 | RW | UNK | Reserved, UNK/SBZP. |
| 7:4 | RW | UNK | Specifies which of the 16 domains, D15-D0, was being accessed when a data fault occurred.  For permission faults that generate Data Abort exception, this field is Unknown. ARMv7 deprecates any use of the domain field in the DFSR. |
| 3:0 | RW | UNK | Fault Status bits. This field indicates the type of exception generated, any encoding not listed is reserved:  0b00001 Alignment fault.  0b00010 Debug event.  0b00011 Access flag fault, section.  0b00100 Instruction cache maintenance fault.  0b00101 Translation fault, section.  0b00110 Access flag fault, page.  0b00111 Translation fault, page.  0b01000 Synchronous external abort, non-translation.  0b01001 Domain fault, section.  0b01011 Domain fault, page.  0b01100 Synchronous external abort on translation table walk, 1st level.  0b01101 Permission fault, section.  0b01110 Synchronous external abort on translation table walk, 2nd level.  0b01111 Permission fault, page.  0b10110 Asynchronous external abort |

Data Fault Status Register

Table .DFSR bit assignments for Long-descriptor translation table format

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:14 | -- | -- | Reserved, UNK/SBZP. |
| 13 | RW | UNK | Cache maintenance fault. For synchronous faults, this bit indicates whether a cache maintenance operation generated the fault:  0 Abort not caused by a cache maintenance operation.  1 Abort caused by a cache maintenance operation |
| 12 | RW | UNK | External abort type. This field indicates whether an AXI Decode or Slave error caused an abort:  0 External abort marked as DECERR.  1 External abort marked as SLVERR.  For aborts other than external aborts this bit always returns 0. |
| 11 | RW | UNK | Write not Read bit. This field indicates whether the abort was caused by a write or a read access:  0 Abort caused by a read access.  1 Abort caused by a write access.  For faults on CP15 cache maintenance operations, including the VA to PA translation operations, this bit always returns a value of 1. |
| 10 | RW | UNK | Reserved, UNK/SBZP |
| 9 | RW | UNK | RAO. |
| 8:6 | RW | UNK | Reserved, UNK/SBZP. |
| 5:0 | RW | UNK | Fault Status bits. This field indicates the type of exception generated. Any encoding not listed is reserved.  0b0001LL Translation fault, LL bits indicate level.  0b0010LL Access fault flag, LL bits indicate level.  0b0011LL Permission fault, LL bits indicate level.  0b010000 Synchronous external abort.  0b010001 Asynchronous external abort.  0b0101LL Synchronous external abort on translation table walk, LL bits indicate level.  0b100001 Alignment fault.  0b100010 Debug event. |

Instruction Fault Status Register

Table .IFSR bit assignments for short-descriptor translation table format

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset value** | **Description** |
| 31:13 | -- | UNK | Reserved, UNK/SBZP. |
| 12 | RW | UNK | External abort type. This field indicates whether an AXI Decode or Slave error caused an abort:  0 External abort marked as DECERR.  1 External abort marked as SLVERR.  For aborts other than external aborts this bit always returns 0. |
| 11 | RW | UNK | Reserved, UNK/SBZP. |
| 10 | RW | UNK | Part of the Fault Status field. See bits 3:0 in this table |
| 9 | RW | UNK | RAZ |
| 8:4 | RW | UNK | Reserved, UNK/SBZP. |
| 3:0 | RW | UNK | Fault Status bits. This field indicates the type of exception generated. Any encoding not listed is reserved.  0b00010 Debug event  0b00011 Access flag fault, section.  0b00101 Translation fault, section.  0b00110 Access flag fault, page.  0b00111 Translation fault, page.  0b01000 Synchronous external abort, non-translation.  0b01001 Domain fault, section.  0b01011 Domain fault, page.  0b01100 Synchronous external abort on translation table walk, 1st level.  0b01101 Permission Fault, Section.  0b01110 Synchronous external abort on translation table walk, 2nd Level.  0b01111 Permission fault, page |

Instruction Fault Status Register

Table . IFSR bit assignments for Long-descriptor translation table format

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:13 | -- | UNK | Reserved, UNK/SBZP. |
| 12 | RW | UNK | External abort type. This field indicates whether an AXI Decode or Slave error caused an abort:  0 External abort marked as DECERR  1 External abort marked as SLVERR.  For aborts other than external aborts this bit always returns 0 |
| 11:10 | RW | UNK | Reserved, UNK/SBZP. |
| 9 | RW | UNK | RAO |
| 8:6 | RW | UNK | Reserved, UNK/SBZP. |
| 5:0 | RW | UNK | Fault Status bits. This field indicates the type of exception generated. Any encoding not listed is reserved.  0b0001LL Translation fault, LL bits indicate level.  0b0010LL Access fault flag, LL bits indicate level.  0b0011LL Permission fault, LL bits indicate level.  0b010000 Synchronous external abort.  0b100010 Debug event |

Hyp Syndrome Register

Table . HSR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:26 | RW | UNK | Exception class. The exception class for the exception that is taken to Hyp mode. When zero, this field indicates that the reason for the exception is not known. In this case, the other fields in this register are UNKNOWN.  Otherwise, the field holds the exception class for the exception. See the ARM Architecture Reference Manual for more information |
| 25 | RW | UNK | Instruction length. Indicates the size of the instruction that has been trapped to Hyp mode:  0 16-bit instructions.  1 32-bit instruction.  This field is not valid for:  • Prefetch Aborts.  • Data Aborts that do not have ISS information, or for which the ISS is not valid. In these cases the field is UNK/SBZP. |
| 24:0 | RW | UNK | Instruction specific syndrome. See the ARM Architecture Reference Manual for more information. The interpretation of this field depends on the value of the EC field. See Encoding of ISS24:20 when HSR31:30 is 0b00. |

L2 Control Register

Table . L2CTLR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:26 | -- | -- | Reserved, RAZ/WI |
| 25:24 | RO | 0x0 | Number of processors present:  0b00 One processor, Processor 0.  0b01 Two processors, Processor 0 and Processor 1.  0b10 Three processors, Processor 0, Processor 1, and Processor 2.  0b11 Four processors, Processor 0, Processor 1, Processor 2, and Processor 3.  These bits are read-only and the reset value of this field is set to the number of processors present in the configuration. |
| 23 | RO | 0x0 | Interrupt controller:  0 Interrupt Controller not present.  1 Interrupt Controller present |
| 22:1 | -- | -- | Reserved, RAZ/WI. |
| 0 | RO | 0x0 | L2 data RAM latency:  0 2 cycles.  1 3 cycles. |

L2 Extended Control Register

Table . L2ECTLR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:30 | -- | -- | Reserved, RAZ/WI |
| 29 | RW | 0x0 | AXI asynchronous error indication:  0 No pending AXI asynchronous error. This is the reset value.  1 AXI asynchronous error has occurred.  A write of 0 clears this bit. A write of 1 is ignored. |
| 28:0 | -- | -- | Reserved, RAZ/WI |

Configuration Base Address Register

Table . CBAR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:15 | RO | -a | The primary input PERIPHBASE31:15 determines the reset value |
| 11:8 | -- | -- | Reserved, UNK/SBZP |
| 7:0 | RO | -a | The primary input PERIPHBASE39:32determines the reset value |

Note:

a. The reset value depends on the primary input, PERIPHBASE39:15

Interrupt Controller Type Register

Address: Operational Base + offset (0x004)

Table . GICD\_TYPER bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | -- | -- | Reserved, RAZ. |
| 15:11 | RO | -b | Returns the number of Lockable Shared Peripheral Interrupts(LSPIs) that the Interrupt Controller  contains:  0b11111 31 LSPIs, these are the interrupts of IDs 32-62.  When CFGSDISABLE is asserted, the GIC prevents writes to any register locations that control the operating state of an LSPI. |
| 10 | RO | -b | Indicates whether the GIC implements the Security Extensions. This bit always returns a value of 1, indicating that the Security Extensions are implemented. |
| 9:8 | -- | -- | Reserved, RAZ |
| 7:5 | RO | -b | Indicates the number of implemented processors:  0b000 The Cortex-A7 MPCore configuration contains one processor.  0b001 The Cortex-A7 MPCore configuration contains two processors.  0b010 The Cortex-A7 MPCore configuration contains three processors.  0b011 The Cortex-A7 MPCore configuration contains four processors.  All other values are reserved for future expansions. |
| 4:0 | RO | -b | Indicates the number of interrupts that the GIC supports:  0b00000 Up to 32 interrupts a, no external interrupt lines.  0b00001 Up to 64 interrupts, 32 external interrupt lines.  0b00010 Up to 96 interrupts, 64 external interrupt lines.  .  .  .  0b01111 Up to 512 interrupts, 480 external interrupt lines.  All other values are reserved for future expansions. |

Note:

a. The Distributor always uses interrupts of IDs 0 to 31 to control any SGIs and PPIs that the GIC might contain

b. IMPLEMENTATION DEFINED

Distributor Implementer Identification Register

Address: Operational Base + offset (0x008)

Table . GICD\_IIDR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RO | 0x01 | Indicates the product ID of the GIC:  0x01 Cortex-A7 |
| 23:20 | RO | -- | Reserved, RAZ |
| 19:16 | RO | 0x01 | Indicates the major revision number of the GIC:  0x0 Variant number |
| 15:12 | RO | 0x01 | Indicates the minor revision number of the GIC:  0x1 Revision number |
| 11:0 | RO | 0x43B | Indicates the implementer:  0x43B ARM implementation |

Private Peripheral Interrupt Status Register

Address: Operational Base + offset (0xD00)

Table . GICD\_PPISR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | RO | -- | Reserved, RAZ |
| 15:9 | RO | 0x0 | Assert when the PPI 6:0inputs to the Distributor are asserted:  PPI6 Virtual Maintenance Interrupt.  PPI5 Hypervisor timer event.  PPI4 Virtual timer event.  PPI3 nIRQ.  PPI2 Non-secure physical timer event.  PPI1 Secure physical timer event.  PPI0 nFIQ.  Note: These bits return the actual status of the PPI6:0 signals. The GICD\_ISPR and GICD\_ICPR can also provide the PPI6:0status, although you can write to these registers, they might not contain the true status of the PPI input signals. |
| 8:0 | RO | -- | Reserved, RAZ |

Shared Peripheral Interrupt Status Registers

Address: Operational Base + offset (0xD04-0xD3C)

Table . GICD\_SPISRn bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | Returns the status of the IRQS479:0signals on the Distributor. For each bit:  0 IRQS is LOW.  1 IRQS is HIGH.  Note:  • The IRQS that a bit refers to depends on its bit position and the base address offset of the GICD\_SPISRn.  • These bits return the actual status of the IRQS479:0 signals. The GICD\_ISPENDRn and GICD\_ICPENDRn can also provide the IRQS479:0status, although you can write to  these registers, they might not contain the actual status of the IRQS479:0 signals |

CPU Interface Identification Register

Address: Operational Base + offset (0x00FC)

Table . GICC\_IIDR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:20 | RO | 0x010 | Identifies the product ID:  0x010 Cortex-A7 MPCore product ID. |
| 19:16 | RO | 0x2 | Identifies the architecture version of the GIC:  0x2 Version 2.0. |
| 15:12 | RO | 0x1 | Identifies the revision number for the CPU interface:  0x1 Revision r0p1. |
| 11:0 | RO | 0x43B | Contains the JEP106 code of the company that implemented the CPU interface. For an ARM implementation, these values are:  Bits 11:8 = 0x4 The JEP106 continuation code of the implementer.  Bit 7 Always 0.  Bits 6:0 = 0x3B The JEP106 identity code of the implementer. |

VGIC Type Register

Address: Operational Base + offset (0x004)

Table .GICH\_VTR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:29 | RO | 0x4 | Indicates the number of priority bits implemented, minus one:  0x4 5 bits of priority and 32 priority levels |
| 28:86 | RO | 0x0 | Indicates the number of pre-emption bits implemented, minus one:  0x4 5 bits of pre-emption and 32 pre-emption levels |
| 25:6 | RO | -- | Reserved, RAZ |
| 5:0 | RO | 0x3 | Indicates the number of implemented List Registers, minus one:  0x3 4 List Registers |

Debug Identification Register

Address: Operational Base + offset (0x000)

Table . DBGDIDR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:28 | RO | Indicates the number of Watchpoint Register Pairs(WRPs) implemented:  0x3 The processor implements 4 WRPs |
| 27:24 | RO | Indicates the number of Breakpoint Register Pairs(BRPs) implemented:  0x5 The processor implements 6 BRPs |
| 23:20 | RO | Indicates the number of BRPs that can be used for Context ID comparison:  0x1 The processor implements 2 breakpoints with Context ID comparison |
| 19:16 | RO | Indicates the Debug architecture version:  0x5 The processor implements ARMv7.1 Debug architecture |
| 15 | RO | Indicates if the Debug Device ID Register(DBGDEVID) is implemented:  1 DBGDEVID is implemented |
| 14 | RO | Indicates if the SecureUser Halting Debug is implemented:  1 Secure User halting debug is not implemented |
| 13 | RO | Indicates if the Program Counter Sampling Register (DBGPCSR) implemented as register 33:  1 DBGPCSR is implemented as register 33 |
| 12 | RO | Security Extensions implemented bit:  1 The processor implementsSecurity Extensions |
| 11:8 | RO | Reserved. |
| 7:4 | RO | This field indicates the variant number of the processor. This number is incremented on functional changes.  The value matches bits 23:20 ofthe CP15 Main ID Register. |
| 3:0 | RO | This field indicates the revision number of the processor. This number is incremented on bug fixes. The value matches bits 3:0 of the CP15 Main ID Register. |

Program Counter Sampling Register

Address: Operational Base + offset (0x084)

Table .DBGPCSR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:1 | RO | Program Counter sample value. The sampled value of bits 31:1 of the PC. The sampled value is either the virtual address of an instruction, or the virtual address of an instruction address plus an offset that depends on the processor instruction set state.  DBGDEVID1.PCSROffset indicates whether an offset is applied to the sampled addresses. |
| 0 | RO | This bit indicates whether the sampled address is an ARM instruction, or a Thumb or ThumbEE instruction:  0 If DBGPCSR1 is 0, the sampled address is an ARM instruction  1 The sampled address is a Thumb or ThumbEE instruction  If T is 0 then DBGPCSR1 is 0, (DBGPCSR31:2 <<2) is the address of the sampled ARM instruction  If T is 1, (DBGPCSR 31:1 << 1) is the address of the sampled Thumb or ThumbEE instruction. |

Debug Run Control Register

Address: Operational Base + offset (0x090)

Table .DBGDRCR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:5 | WO | Reserved, SBZ |
| 4 | WO | Cancel Bus Requests Request. The actions on writing to this bit are:  0 No action  1 Request cancel of pending accesses |
| 3 | WO | Clear Sticky Pipeline Advance bit. This bit sets the DBGDSCR. PipeAdv bit to 0. The actions on writing to this bit are:  0 No action  1 Sets the DBGDSCR. PipeAdv bit to 0  When the processor is powered down, it is UNPREDICTABLE whether a write of 1 to this bit sets DBGDSCR. PipeAdv to 0. |
| 2 | WO | Clear Sticky Exceptions bits. This bit sets the DBGDSCR sticky exceptions bits to 0. The actions on writing to this bit are:  0 No action  1 Sets the DBGDSCR8:6 to 0b000  See the ARM Architecture Reference Manual for more information on the DBGDSCR. |
| 1 | WO | Restart request. The actions on writing to this bit are:  0 No action  1 Request exit from Debug state  Writing 1 to this bit requests that the processor exits Debug state. This request is held until the processor exits Debug state. After the request has been made, the debugger can poll the DBGDSCR.RESTARTED bit until it reads as 1.  The processor ignores writes to this bit if it is in Non-debug state. |
| 0 | WO | Halt request. The actions on writing to this bit are:  0 No action  1 Request entry to Debug state  Writing 1 to this bit requests that the processor enters Debug state. This request is held until the processor enters Debug state.  After the request has been made, the debugger can poll the DBGDSCR.HALTED bit until it reads 1.  The processor ignores writes to this bit if it is already in Debug state. |

Debug External Auxiliary Control Register

Address: Operational Base + offset (0x094)

Table . DBGEACR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:4 | RW | Reserved, RAZ/WI. |
| 3 | RW | Read-only status bit that reflects the current reset state of the debug logic in the CPU power domain:  0 Debug logic in CPU power domain is not in reset state  1 Debug logic in CPU power domain is currently in reset state |
| 2:0 | RW | Reserved, RAZ/WI. |

Debug Identification Register

Address: Operational Base + offset (0x100-0x114)

Table .DBGBVR bit assignments when register is used for address comparison

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:2 | RW | This field indicates bits31:2 of the address value for comparison |
| 1:0 | RW | This field must be written as 0b00, otherwise the generation of breakpoint debug events by this breakpoint is UNPREDICTABLE |

Table .DBGBVR bit assignments when register is used for Context ID comparison

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:0 | RW | This field indicates bits31:0 of the Context ID value for comparison |

Breakpoint Control Registers

Address: Operational Base + offset (0x140-0x154)

Table .DBGBCR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:29 | RW | Reserved. |
| 28:24 | RW | Address mask. The processor does not support address range masking. |
| 23:20 | RW | Breakpoint Type. This field controls the behavior of Breakpoint debug event generation. This includes the meaning of the value held in the associated DBGBVR, indicating whether it is an instruction address match or mismatch or a Context match. It also controls whether the breakpoint is linked to another breakpoint.  See the ARM Architecture Reference Manual for the meaning of bits 23:20. |
| 19:16 | RW | Linked Breakpoint Number. If this breakpoint is programmed for Linked instruction address match or mismatch then this field must be programmed with the number of the breakpoint that holds the Context match to be used in the combined instruction address and Context comparison. Otherwise, this field must be programmed to 0b0000.  Reading this register returns an UNKNOWN value for this field, and the generation of debug events is UNPREDICTABLE, if either:  • This breakpoint is not programmed for Linked instruction address match or mismatch and this field is not programmed to 0b0000.  • This breakpoint is programmed for Linked instruction address match or mismatch and the breakpoint indicated by this field does not support Context matching or is not programmed for Linked Context ID match.  See the ARM Architecture Reference Manual for more information. |
| 15:14 | RW | Security State Control. This field enables the watch point to be conditional on the security state of the processor.  This field is used with the Hyp Mode Control (HMC), and Privileged Mode Control (PMC), fields. See the ARM Architecture Reference Manual for possible values of the fields, and the mode and security states that can be tested.  This field must be programmed to b00if DBGBCR.BT is programmed for Linked Context match. If this is not done, the generation of debug events by this breakpoint is UNPREDICTABLE.  Note: When this field is set to a value other than b00, the SSC field controls the processor security state in which the access matches, not the required security attribute of the access. |
| 13 | RW | Hyp Mode Control. This field is used with the SSC and PMC fields. See the ARM Architecture Reference Manual for possible values of the fields, and the access modes and security states that can be tested.  This field must be programmed to 0 if DBGBCR.BT is programmed for Linked Context match. If this is not done, the generation of debug events by this breakpoint is UNPREDICTABLE. |
| 12:9 | RW | Reserved |
| 8:5 | RW | Byte Address Select. This field enables match or mismatch comparisons on only certain bytes of the word address held in the DBGBVR. The operation of this field depends also on:  • The breakpoint type field being programmed for instruction address match or mismatch.  • The Address range mask field being programmed to 0b00000, no mask.  • The instruction set state of the processor, indicated by the CPSR.J and CPSR.T bits.  This field must be programmed to 0b1111if either:  • The DBGBVR.BT is programmed for Linked or Unlinked Context ID match.  • DBGBCR.MASK is programmed to a value other than 0b00000.  If this is not done, the generation of Breakpoint debug events is UNPREDICTABLE. |
| 4:3 | RW | Reserved. |
| 2:1 | RW | Privileged Mode Control. This field enables breakpoint matching conditional on the mode of the processor:  This field is used with the SSC and HMC fields. See the ARM Architecture Reference Manual for possible values of the fields, and the mode and security states that can be tested.  This field must be programmed to 0b11if DBGBCR.BT is programmed for Linked Context ID match. If this is not done, the generation of debug events by this breakpoint is UNPREDICTABLE.  Note: Bits 2:1 has no effect for accesses made in Hyp mode. |
| 0 | RW | Breakpoint Enable. This bit enables the breakpoint:  0 Breakpoint disabled  1 Breakpoint enabled  A breakpoint never generates a debug event when it is disabled.  Note: The value of the DBGBCR.E bit is UNKNOWN on reset. A debugger must ensure that DBGBCR.E has a defined value before it programs DBGDSCR15:14 to enable debug. |

Watchpoint Value Registers

Address: Operational Base + offset (0x180-0x18C)

Table .DBGWVR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:2 | RW | This field indicates bits31:2 of the address value for comparison |
| 1:0 | RW | RAZ on reads and SBZP for writes |

Watchpoint Control Registers

Address: Operational Base + offset (0x1C0-0x1CC)

Table . DBGWCR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:29 | RW | Reserved. |
| 28:24 | RW | Address mask. The processor supports watch point address masking. This field can be used to set a watch point on a range of addresses by masking lower order address bits out of the watch point comparison. The value of this field is the number of low order bits of the address that are masked off, except that values of 1 and 2 are reserved.  Therefore, the meaning of Watch point Address range mask values are:  0b00000 No mask  0b00001 Reserved  0b00010 Reserved  0b00011 0x00000007 mask for data address, three bits masked  0b00100 0x0000000F mask for data address, four bits masked  0b00101 0x0000001F mask for data address, five bits masked  ….  0b11111 0x7FFFFFFF mask for data address, 31 bits masked.  See the ARM Architecture Reference Manual for the meanings of watch point address range mask values |
| 23:21 | RW | Reserved. |
| 20 | RW | Watch point Type. This bit is set to 1 to link the watch point to a breakpoint to create a linked watch point that  requires both data address matching and Context matching:  0 Linking disabled  1 Linking enabled  When this bit is set to 1 the linked breakpoint number field indicates the breakpoint to which this watch point is linked. See the ARM Architecture Reference Manual for more information. |
| 19:16 | RW | Linked Breakpoint Number. If this watch point is programmed with watch point type set to linked, then this field must be programmed with the number of the breakpoint that defines the Context match to be combined with data address comparison. Otherwise, this field must be programmed to 0b0000.  Reading this register returns an UNKNOWN value for this field, and the generation of Watch point debug events is UNPREDICTABLE, if either:  • This watch point does not have linking enabled and this field is not programmed to 0b0000.  • This watch point has linking enabled and the breakpoint indicated by this field does not support Context matching, is not programmed for Context matching, or does not exist.  See the ARM Architecture Reference Manual for more information. |
| 15:14 | RW | Security State Control. This field enables the watch point to be conditional on the security state of the processor. This field is used with the Hyp Mode Control (HMC) and Privileged Access Control (PAC) fields.  See the ARM Architecture Reference Manual for possible values of the fields, and the access modes and security states that can be tested. |
| 13 | RW | Hyp Mode Control. This field is used with the Security State Control (SSC) and PAC fields. The value of DBGWCR.PAC has no effect for accesses made in Hyp mode.  See the ARM Architecture Reference Manual for possible values of the fields, and the access modes and security states that can be tested. |
| 12:5 | RW | Byte Address Select. The processor implements an 8-bit Byte address select field, DBGWCR 12:5.  A DBGWVR is programmed with a word-aligned address. This field enables the watch point to hit only if certain bytes of the addressed word are accessed. The watch point hits if an access hits any byte being watched, even if:  • The access size is larger than the size of the region being watched.  • The access is unaligned, and the base address of the access is not in the same word of memory as the address in the DBGWVR.  See the ARM Architecture Reference Manual for more information. |
| 4:3 | RW | Load/store access control. This field enables watch point matching conditional on the type of access being made:  0b00 Reserved  0b01 Match on any load, Load-Exclusive, or swap  0b10 Match on any store, Store-Exclusive or swap  0b11 Match on either type of access |
| 2:1 | RW | Privileged Access Control. This field enables watch point matching conditional on the mode of the processor.  This field is used with the SSC and HMC fields.  See the ARM Architecture Reference Manual for possible values of the fields, and the access modes and security states that can be tested. |
| 0 | RW | Watch point Enable. This bit enables the watch point:  0 Watch point disabled  1 Watch point enabled  A watch point never generates a Watch point debug event when it is disabled.  For more information about possible watch points values, see Watch point Value Registers  Note: The value of DBGWCR.E is UNKNOWN on reset. A debugger must ensure that DBGWCR.E has a defined value before it programs DBGDSCR 15:14 to enable debug. |

Debug ROM Address Register

Table . DBGDRAR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 64:40 | RO | Reserved |
| 39:32 | RO | Bits 39:32 of the ROM table physical address. Bits11:0 of the address are zero.  If DBGDRAR. Valid is zero the value of this field is UNKNOWN. |
| 31:12 | RO | Bits 31:12 of the ROM table physical address. Bits 11:0 of the address are zero.  If DBGDRAR. Valid is zero the value of this field is UNKNOWN. |
| 11:2 | RO | Reserved. |
| 1:0 | RO | Valid bits. This field indicates whether the ROM table address is valid:  0b00 ROM table address is not valid  0b11 ROM table address is valid  Note: ROMADDRV must be set to 1 if ROMADDR39:12is set to a valid value |

Breakpoint Extended Value Registers

Address: Operational Base + offset ( 0x250-0x254)

Table . DBGBXVR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:8 | RW | UNK/SBZP |
| 7:0 | RW | VMID value. Used to compare with the Virtual Machine ID(VMID) field, held in the Virtualization Translation  Table Base Register (VTTBR). See the ARM Architecture Reference Manual for more information. |

OS Lock Access Register

Address: Operational Base + offset ( 0x300)

Table . DBGOSLAR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:0 | WO | Writing the key value 0xC5ACCE55to this field locks the debug registers.  Writing any other value to this register unlocks the debug registers if they are locked. See the ARM Architecture Reference Manual for more information. |

OS Lock Status Register

Address: Operational Base + offset (Operational Base + offset (0x304)

Table . DBGOSLSR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:4 | RO | Reserved, UNK. |
| 3 | RO | OS Lock Model implemented bit. This field identifies the form of OS Save and Restore mechanism implemented:  0b10 The processor implements the OS Lock Model but does not implement DBGOSSRR  Note: This field splits across the two non-contiguous bits in the register. |
| 2 | RO | This bit is always RAZ. It indicates that a 32-bit access is needed to write the key to the OS Lock Access Register. |
| 1 | RO | This bit indicates the status of the OS Lock:  0 Lock not set  1 Lock set  The OS Lock is set or cleared by writing to the DBGOSLAR, see OS Lock Access Register. The OS Lock is set to 1 on a core power up reset.  Setting the OS Lock restricts access to Debug registers. See the ARM Architecture Reference Manual for more information. |
| 0 | RO | OS Lock Model implemented bit. This field identifies the form of OS Save and Restore mechanism implemented:  0b10 The processor implements the OS Lock Model but does not implement DBGOSSRR  Note: This field splits across the two non-contiguous bits in the register. |

Device Power-down and Reset Control Register

Address: Operational Base + offset (Operational Base + offset (0x310)

Table . DBGPRCR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:4 | RW | Reserved, UNK/SBZP. |
| 3 | RW | Power-up request bit. This bit enables a debugger to request that the power controller powers up the processor, enabling access to debug registers in the processor power domain:  0 DBGPWRUPREQ is LOW, this is the reset value.  1 DBGPWRUPREQ is HIGH. This bit is only defined for the memory-mapped and external debug interfaces.  Note: This bit never affects system power-up, because when implemented it resets to 0.  For accesses to DBGPRCR from CP14, this bit is UNK/SBZP. See the ARM Architecture Reference Manual for more information. |
| 2 | RW | Hold reset bit. Writing 1 to this bit means the non-debug logic of the processor is held in reset after a  power-up or warm reset:   1. Do not hold the non-debug logic reset on power-up or warm reset   1 Hold the non-debug logic of the processor in reset on power-up or warm reset  The processor is held in this state until this bit is set to 0  Note: This bit never affects system power-up, because when implemented it resets to 0.  For accesses to DBGPRCR from CP14, this bit is UNK/SBZP.  See the ARM Architecture Reference Manual for more information. |
| 1 | RW | Reset request bit. Writing 1 to this bit issues a request for a warm reset:  0 No action  1 Request internal reset using the memory mapped interface or CP14  Reads from this bit are UNKNOWN, and writes to this bit from the memory-mapped or external debug interface are ignored when any of the following apply:  • The core power domain is off.  • DBGPRSR.DLK, OS Double Lock status bit, is set to 1.  • For the external debug interface, the OS lock is set.  See the ARM Architecture Reference Manual for more information |
| 0 | RW | Hold power-up request bit. When set to 1, the DBGNOPWRDWN output signal is HIGH. This output is connected to the system power controller and is interpreted as a request to operate in emulate mode. In this mode, the processor that includes ETM are not actually powered down when requested by software or hardware handshakes:  0 DBGNOPWRDWN is LOW. This is the reset value.  1 DBGNOPWRDWN is HIGH.  This bit is UNKNOWN on reads and ignores writes when any of the following apply:  • The core power domain is powered down. If the CORENPDRQ bit is 1, it loses this value through the power down.  • DBGPRSR.DLK, OS Double Lock status bit is set to 1.  • For the external debug interface, the OS Lock is set. |

Debug Self Address Offset Register

Table . DBGDSAR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 63:40 | RO | Sign extension. |
| 39:32 | RO | Bits 39:32 of the two’s complement offset from the base address defined by DBGDRAR to the physical address where the debug registers are mapped. Bits 11:0 of the address are zero. See Debug ROM Address Register.  If the DBGDSAR. Valid is zero, is zero the value of this field is UNKNOWN |
| 31:12 | RO | Bits 31:12 of the two’s complement offset from the base address defined by DBGDRAR to the physical address where the debug registers are mapped. Bits 11:0 of the address are zero. See Debug ROM Address Register  If DBGDSAR. Valid is zero, the value of this field is UNKNOWN. |
| 11:2 | RO | Reserved |
| 1:0 | RO | Valid bit. This field indicates whether the debug self address offset is valid:  0b00 Offset is not valid  0b11 Offset is valid |

Integration Miscellaneous Signals Register

Address: Operational Base + offset ( 0xEF8)

Table . DBGITMISCOUT bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:10 | WO | Reserved, SBZP |
| 9 | WO | Set value of the DBGRESTARTED output pin. Handshake for DBGRESTART that is the same as the DSCR 1 bit. The reset value is 0. |
| 8:5 | WO | Reserved, SBZP |
| 4 | WO | Set value of PMUIRQ output pin. When this bit is set to 1, the corresponding nPMUIRQ signal is cleared to 0. The reset value is 1. |
| 3:1 | WO | Reserved, SBZP |
| 0 | WO | Set value of the DBGACK output pin |

Integration Miscellaneous Signals Input Register

Address: Operational Base + offset ( 0xEFC)

Table . DBGITMISCIN bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:3 | RO | Reserved, RAZ. |
| 2 | RO | Read value of nFIQ input pin. |
| 1 | RO | Read value of nIRQ input pin. |
| 0 | RO | Read value of EDBGRQ input pin |

Integration Mode Control Register

Address: Operational Base + offset (0xF00)

Table . DBGITCTRL bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:1 | RW | Reserved. |
| 0 | RW | When this bit is set to 1, the device reverts to an integration mode to enable integration testing  or topology detection:  0 Normal operation  1 Integration mode enabled |

Claim Tag Set Register

Address: Operational Base + offset (0xFA0)

Table . DBGCLAIMSET bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:8 | RW | Reserved |
| 7:0 | RW | CLAIM bits. Writing a 1 to one of these bits sets the corresponding CLAIM bit to 1. A single write operation can set multiple bits to 1. The CLAIM bits do not have any specific functionality. ARM expects the usage model to be that an external debugger and a debug monitor can set specific bits to 1 to claim the corresponding debug resources.  This field is RAO.  See Claim Tag Clear Register for information on how to:  • Clear CLAIM bits to 0.  • Read the current values of the CLAIM bits. |

Claim Tag Clear Register

Address: Operational Base + offset (0xFA4)

Table . DBGCLAIMCLR bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:8 | RW | Reserved, RAZ/SBZP. |
| 7:0 | RW | CLAIM bits. Writing a 1 to one of these bits sets the corresponding CLAIM bit to 0. A single write operation can clear multiple bits to 0. Writing 0 to one of these bits has no effect.  Reading the register returns the current values of these bits. The debug logic reset value of these bits is 0.  For more information about the CLAIM bits and how they might be used, see Claim Tag Set Register |

Debug Device ID Register 1

Address: Operational Base + offset (0xFC4)

Table . DBGDEVID1 bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:4 | RO | Reserved |
| 3:0 | RO | Defines the offset applied to DBGPCSR samples:  0b0001 DBGPCSR samples have no offset applied |

Debug Device ID Register

Address: Operational Base + offset (0xFC8)

Table . DBGDEVID bit assignments

|  |  |  |
| --- | --- | --- |
| **Bit** | **Type** | **Description** |
| 31:28 | RO | Indicates the level of support for the Context ID matching breakpoint masking capability:  0b0000 Context ID masking is not implemented |
| 27:24 | RO | Specifies support for the Debug External Auxiliary Control Register. See Debug External Auxiliary Control Register on page 10-12:  0b0001 The processor supports Debug External Auxiliary Control Register |
| 23:20 | RO | Specifies support for the Debug OS Double Lock Register:  0b0001 The processor supports Debug OS Double-lock Register |
| 19:16 | RO | Specifies the implementation of the Virtualization Extensions to the Debug architecture:  0b0001 The processor implements the Virtualization Extensions to the Debug architecture |
| 15:12 | RO | Defines the form of the vector catch event implemented:  0b0000 The processor implements address matching form of vector catch |
| 11:8 | RO | Indicates the level of support for the Immediate Virtual Address (IVA) matching breakpoint masking capability:  0b1111 Breakpoint address masking not implemented DBGBCRn 28:24 are UNK/SBZP |
| 7:4 | RO | Indicates the level of support for the DVA matching watchpoint masking capability:  0b0001 Watch point address mask implemented |
| 3:0 | RO | Indicates the level of support for Program Counter sampling using debug registers 40, 41, and 42:  0b0011 DBGPCSR,DBGCIDSR and DBGVIDSR are implemented as debug registers 40, 41, and 42 |

Debug Peripheral Identification Registers

Address: Operational Base + offset ( 0xFD0~ 0xFEC)

Table . Debug Peripheral ID 4/5/6/7/0/1/2/3

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | -- | Reserved |
| 7:0 a | RO | 0x04 | Debug Peripheral ID 4 |
| 7:0 | RO | 0x00 | Debug Peripheral ID 5 |
| 7:0 | RO | 0x00 | Debug Peripheral ID 6 |
| 7:0 | RO | 0x00 | Debug Peripheral ID 7 |
| 7:0 | RO | 0x07 | Debug Peripheral ID 0 |
| 7:0 | RO | 0xBC | Debug Peripheral ID 1 |
| 7:0 | RO | 0x5B | Debug Peripheral ID 2 b |
| 7:0 | RO | 0x00 | Debug Peripheral ID 3 |

Note:

a. Only bits 7:0 of each Debug Peripheral ID Register are used, with bits 31:8 reserved. Together, the eight Debug Peripheral ID Registers define a single 64-bit Debug Peripheral ID.

b. Bits 7:4 of this value match the revision field in the Debug Identification Register

Debug Component Identification Registers

Address: Operational Base + offset ( 0xFF0~ 0xFFC)

Table . Summary of the Component Identification Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | -- | Reserved |
| 7:0 | RO | 0x0D | Debug Peripheral ID 0 |
| 7:0 | RO | 0x90 | Debug Peripheral ID 1 |
| 7:0 | RO | 0x05 | Debug Peripheral ID 2 |
| 7:0 | RO | 0xB1 | Debug Peripheral ID 3 |

Performance Monitor Control Register

Address: Operational Base + offset (0xE04)

Table . PMCR bit assignments

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RW | 0x41 | Implementer code.  0x41 ARM.  This is a read-only field. |
| 23:16 | RW | 0x07 | Identification code.  0x07 Cortex-A7 MPCore identification code.  This is a read-only field. |
| 15:11 | RW | 0x4 | Number of event counters. In Secure state and Hyp mode, this field return 0x4that indicates the number of counters implemented.  In Non-secure modes other than Hyp mode, this field reads the value of HDCR.HPMN. See Hyp Debug Control Register.  This is a read-only field |
| 10:6 | RW | -- | Reserved, UNK/SBZP. |
| 5 | RW | 0 | Disable cycle counter, PMCCNTR, in regions of software when prohibited:  0 Count is enabled in prohibited regions. This is the reset value  1 Count is disabled in prohibited regions. This bit is read/write |
| 4 | RW | 0 | Export enable. This bit permits events to be exported to another debug device, such as a trace macro cell, over  an event bus:  0 Export of events is disabled. This is the reset value.  1 Export of events is enabled. This bit is read/write. |
| 3 | RW | 0 | Clock divider:  0 When enabled, PMCCNTR counts every clock cycle. This is the reset value.  1 When enabled, PMCCNTR counts once every 64 clock cycles.  This bit is read/write. |
| 2 | WO | 0 | Clock counter reset:  0 No action. This is the reset value.  1 Reset PMCCNTR to 0.  This bit is write-only, and always RAZ. |
| 1 | WO | 0 | Event counter reset:  0 No action. This is the reset value.  1 Reset all event counters, not including PMCCNTR, to 0.  In Non-secure modes other than Hyp mode, writing a 1 to this bit does not reset event counters that the  HDCR.HPMN field reserves for Hyp mode use. See Hyp Debug Control Register on page 4-68.  In Secure state and Hyp mode, writing a 1 to this bit resets all event counters.  This bit is write-only, and always RAZ. |
| 0 | RW | 0 | Enable bit. Performance monitor overflow IRQs are only signaled when the enable bit is set to 1.  0 All counters, including PMCCNTR, are disabled. This is the reset value.  1 All counters are enabled.  This bit is read/write. |

Performance Monitors Peripheral Identification Registers

Address: Operational Base + offset ( 0xFD0~ 0xFEC)

Table . The performance monitors peripheral identification registers 4/5/6/7/0/1/2/3

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | -- | Reserved |
| 7:0 a | RO | 0x04 | Performance Monitors Peripheral ID 4 |
| 7:0 | RO | 0x00 | Performance Monitors Peripheral ID 5 |
| 7:0 | RO | 0x00 | Performance Monitors Peripheral ID 6 |
| 7:0 | RO | 0x00 | Performance Monitors Peripheral ID 7 |
| 7:0 | RO | 0xA7 | Performance Monitors Peripheral ID 0 |
| 7:0 | RO | 0xB9 | Performance Monitors Peripheral ID 1 |
| 7:0 | RO | 0x5B | Performance Monitors Peripheral ID 2 b |
| 7:0 | RO | 0x00 | Performance Monitors Peripheral ID 3 |

Note:

a.Only bits 7:0 of each Performance Monitors Peripheral ID Register are used, with bits 31:8 reserved. Together, the eight Performance Monitors Peripheral ID Registers define a single 64-bit Peripheral ID

Performance Monitors Component Identification Registers

Address: Operational Base + offset (0xFF0~ 0xFFC)

Table . Summary of the Component Identification Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | -- | Reserved |
| 7:0 | RO | 0x0D | Performance Monitors Component ID 0 |
| 7:0 | RO | 0x90 | Performance Monitors Component ID 1 |
| 7:0 | RO | 0x05 | Performance Monitors Component ID 2 |
| 7:0 | RO | 0xB1 | Performance Monitors Component ID 3 |

1. System Control Unit
   1. Register Summary

Table 100.SoC register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| CID | 0x003540000 | 0x00F2 A000 | Chip id |
| SMMR | 0x0000C | 0x0000 0000 | System memory mapping register |
| **RCU Control Register** | | | |
| RCR | 0x00100 | 0x0000 0FF1 | RCU control register |
| SSR | 0x00104 | 0x001F FFFE | Software reset register |
| PSR | 0x00108 | 0x00FF FF9F | RCU Peripheral Software reset Register |
| NSR | 0x0010C | 0x0000 FFFF | NOC software reset register |
| RSR | 0x00110 | 0x0000 0000 | RCU Register status |
| **CMU Register** | | | |
| CCR | 0x000200 | 0x0000 0000 | CMU control register |
| CSR | 0x000204 | 0x0000 0000 | CMU status register |
| CPLL\_CR0 | 0x000208 | 0x0130 6400 | CMU CPU pll configuration register0 |
| CPLL\_CR1 | 0x00020C | 0x0100 0000 | CMU CPU pll configuration register1 |
| CPLL\_CR2 | 0x000210 | 0x000F F005 | CMU CPU pll configuration register2 |
| DPLL\_CR0 | 0x000214 | 0x0120 3740 | CMU DDR pll configuration register0 |
| DPLL\_CR1 | 0x000218 | 0x018e 3ab5 | CMU DDR pll configuration register1 |
| DPLL\_CR2 | 0x00021C | 0x000F F005 | CMU DDR pll configuration register2 |
| SPLL\_CR0 | 0x000220 | 0x0120 5340 | CMU system pll configuration register0 |
| SPLL\_CR1 | 0x000224 | 0x0155 5555 | CMU system pll configuration register1 |
| SPLL\_CR2 | 0x000228 | 0x000F F005 | CMU system pll configuration register2 |
| VPLL\_CR0 | 0x00022C | 0x0110 3040 | CMU video pll configuration register0 |
| VPLL\_CR1 | 0x000230 | 0x0100 0000 | CMU video pll configuration register1 |
| VPLL\_CR2 | 0x000234 | 0x000F F005 | CMU video pll configuration register2 |
| CSSR | 0x000238 | 0X0000 0000 | CMU Clock Source Select Register |
| MSSR | 0x00023C | 0X0000 0000 | CMU MUX Select Setting Register |
| SDSR | 0x000240 | 0X3540 1055 | CMU SYS Domain Divider Setting Register |
| DDSR | 0x000244 | 0x0000 0025 | CMU Debug Domain Divider Setting Register |
| ODSR | 0x000248 | 0x2801 0354 | CMU Other Domain Divider Setting Register |
| MHIPR | 0x00024C | 0X0000 0000 | CMU MUX Handshake In-Process Register |
| DHIPR | 0x000250 | 0x0000 0000 | CMU Divider Handshake In-Process Register |
| SYSCGR | 0x000254 | 0x01FF 1F7B | CMU system clock gating register |
| DBGGR | 0x000258 | 0x0000 0007 | CMU DBG Domain Clock Gating Register |
| PIOGR | 0x00025C | 0x0000 000F | CMU peripheral IO clock gating register |
| OGR | 0x000260 | 0x0000 393B | CMU Other Domain Clock Gating Register |
| NGR | 0x000264 | 0x0000 FFFF | CMU NOC Clock Gating Register |
| **IP Control register** | | | |
| ICR0 | 0x000304 | 0x0100 221B | ISRAM Control Register0 |
| ICR1 | 0x000308 | 0x01FF 00FF | ISRAM Control Register1 |
| DCR0 | 0x000310 | 0x0000 0000 | DMAC Control Register0 |
| DCR1 | 0x000314 | 0x0000 0000 | DMAC Control Register1 |
| DDRCR | 0x000318 | 0x0000 0000 | DDR Control Register |
| DDRCSR | 0x00031C | 0x0000 0000 | DDR Control/Status Register |
| CRPTCR | 0x000320 | 0x0000 0000 | Crypto Control Register |
| GCR | 0x000324 | 0x0008 0131 | GMAC Control Register |
| SPCR | 0x000328 | 0x0000 0101 | SPINOR Control Register |
| SDRCSR | 0x00032C | 0x0000 0000 | SDIO Control/status Register |
| CAOR0 | 0x000330 | 0x0000 0000 | CA7 Options register0 |
| CAOR1 | 0x000334 | 0x0000 0000 | CA7 Options register1 |
| CAOR2 | 0x000338 | 0x0000 0000 | CA7 Options register2 |
| CTCR0 | 0x000348 | 0x0000 0178 | CoreSight Trace Control Register |
| CTCR1 | 0x00034C | 0x8000 0003 | CoreSight Control Register |
| CTCR2 | 0x000350 | 0x088C 001D | CoreSight Control Register |
| SPRCR | 0x000354 | 0x0001 6E36 | SPINOR Reset control REgister |
| SDCR1 | 0x000358 | 0x0000 0000 | SDIO Control Register1 |

* 1. Detail Register Description

CID: CHIP-ID Register

Address: SCU Base + offset (0x00000)

Table 101.Chip ID Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:0] | CHIP\_ID | 0x00F2\_A000 | RO | CHIP ID |

SMMR: System Memory Mapping Register

Address: SCU Base + offset (0x0000C)

Table 102.System memory mapping register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [1:0] | Reg\_soc\_remap | 0x0 | RW | SoC booting remap |
| [3:2] | Reg\_noc\_remap | 0x0 | RO | Current NOC routing mapping mode |
| [5:4] | - | 0x0 | RO | Reserved |
| [6] | TestNoC | 0x0 | RW | NoC test mode select |
| [31:7] | - | 0x0 | RO | Reserved |

RCR: RCU Control Register (RCR)

Address: SCU Base + offset (0x00100)

Table 103.RCU control register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [0] | RCUReg\_warm\_en | 0x1 | RW | WARM reset enable bit: WARM reset will be generated only if this bit be set, or a COLD reset will be generated instead. |
| 0x0: Disable WARM reset |
| 0x1: Enable WARM reset |
| [3:1] | - | 0x0 | RO | Reserved |
| [7:4] | RCUReg\_mask\_wdog\_rst | 0xf | RW | Mask watch-dog reset: |
| Note: When watch-dog timer expired, the watch-dog time-out status bit will be set (WDOG Reset Status Register). The ‘Hardware reset’ is the only way to clear watch-dog time-out status |
| 0xf Mask watch-dog reset |
|  |  |
| [11:8] | RCUReg\_mask\_btn\_rst | 0xf | RW | Mask btn reset: |
| 0xf Mask btn reset |
|  |
| [31:12] | - | 0x0 | RO | Reserved |

SSR: RCU System Software Reset Register

Address: SCU Base + offset (0x00104)

Table 104.RCU software reset register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:21] | - | - | - | Reserved |
| [20] | RCUReg\_rom\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for rom. |
| [19] | RCUReg\_sw\_sys\_apb\_rstn | 1'b1 | RW | 0x0: Asserts software reset for system apb. |
| [18] | RCUReg\_uart\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for uart. |
| [17] | RCUReg\_gpio\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for gpio. |
| [16] | RCUReg\_spinor\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for spinor. |
| [15] | RCUReg\_timer\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for timer. |
| [14] | RCUReg\_wdt\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for wdt. |
| [13] | RCUReg\_i2c0\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for i2c0. |
| [12] | RCUReg\_crypto\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for crypto. |
| [11] | RCUReg\_dma\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for dma. |
| [10] | RCUReg\_ddr\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ddr. |
| [9] | RCUReg\_isram\_mreset1n\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for isram mreset1n. |
| [8] | RCUReg\_isram\_mreset0n\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for isram mreset0n. |
| [7] | RCUReg\_isram\_aresetn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for isram aresetn. |
| [6] | RCUReg\_cpu\_nsocdbgreset\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ARM nsocdbgreset. |
| [5] | RCUReg\_cpu\_netmreset\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ARM netmreset. |
| [4] | RCUReg\_cpu\_ndbgreset\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ARM ndbgreset. |
| [3] | RCUReg\_cpu\_ncorereset\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ARM CA7 Core. |
| [2] | RCUReg\_cpu\_nl2reset\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ARM CA7 l2reset. |
| [1] | RCUReg\_cpu\_ncoreporeset\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for ARM CA7 Core. |
| [0] | RCUReg\_soc\_sw\_rst | 1'b0 | RW | 0x1: Asserts whole system software reset (except SCU, PMU) |

PSR: RCU Peripheral Software Reset Register

Address: SCU Base + offset (0x00108)

Table 105.RCU peripheral software reset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | RO | Reserved |
| [23] | RCUReg\_iva\_rstn\_vsi\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [22] | RCUReg\_iva\_rstn\_clf\_2x\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [21] | RCUReg\_iva\_rstn\_clf\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [20] | RCUReg\_iva\_rstn\_cfm\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [19] | RCUReg\_iva\_rstn\_flt\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [18] | RCUReg\_iva\_rstn\_vav\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [17] | RCUReg\_iva\_rstn\_cfg\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [16] | RCUReg\_iva\_rstn\_axi\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for iva. |
| [15] | RCUReg\_sdc\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for iva. |
| [14] | RCUReg\_pioapb\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for pio presetn. |
| [13] | RCUReg\_iomux\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for iomux. |
| [12] | RCUReg\_scu\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for scu. |
| [11] | RCUReg\_pmu\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for pmu. |
| [10] | RCUReg\_pwm\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for pwm. |
| [9] | RCUReg\_spi1\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for spi1. |
| [8] | RCUReg\_spi0\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for spi0. |
| [7] | RCUReg\_i2c1\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for i2c1. |
| [6] | - | 1'b0 | RO | Reserved |
| [5] | - | 1'b0 | RO | Reserved |
| [4] | RCUReg\_cs\_tresetn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for cs tresetn. |
| [3] | RCUReg\_cs\_presetn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for cs presetn. |
| [2] | RCUReg\_cs\_npotrst\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for cs npotrst. |
| [1] | RCUReg\_gmac\_divier\_rstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for gmac divider rstn. |
| [0] | RCUReg\_gmac\_hresetn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for gmac hresetn. |

RSR: RCU Reset Status Register

Address: SCU Base + offset (0x0010C)

Table 106.RCU reset status register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:16] | - | 0x0 | RO | Reserved |
| [15] | RCUReg\_noc\_gmac\_sdc\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for noc gmac. |
| [14] | RCUReg\_noc\_gmac\_prstn\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for noc gmac. |
| [13] | RCUReg\_noc\_iva\_prstn\_sw\_rstn | 1'b1 | RW | 0x1: Asserts software reset for noc iva. |
| [12] | RCUReg\_noc\_iva\_arstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc iva. |
| [11] | RCUReg\_noc\_ddr\_arstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc ddr. |
| [10] | RCUReg\_noc\_pio\_prstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc pio presetn. |
| [9] | RCUReg\_noc\_sys\_prstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc sys presetn. |
| [8] | RCUReg\_noc\_spinor\_prstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc spinor prstn. |
| [7] | RCUReg\_noc\_sys\_hrstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc sys hrstn. |
| [6] | RCUReg\_noc\_dma\_prstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc dma prstn. |
| [5] | RCUReg\_noc\_dma\_arstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc dma arstn. |
| [4] | RCUReg\_noc\_crypto\_arstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc crypto arstn. |
| [3] | RCUReg\_noc\_isram\_prstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc isram prstn. |
| [2] | RCUReg\_noc\_isram\_arstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc isram arstn. |
| [1] | RCUReg\_noc\_cs\_prstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc cs prstn. |
| [0] | RCUReg\_noc\_cpu\_arstn\_sw\_rstn | 1'b1 | RW | 0x0: Asserts software reset for noc cpu arstn. |

RGPR0: RCU General Purpose Register 0

Address: SCU Base + offset (0x00110)

Table 107.RCU General Purpose Register0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:14] | - | - | - | - |
| [13] | RCUReg\_pd\_dbg\_reset | 0x0 | RW1C | 0x1: Indicates that a dbg power domain reset has happened. |
| [12] | RCUReg\_pd\_pio\_reset | 0x0 | RW1C | 0x1: Indicates that a pio power domain reset has happened. |
| [11] | RCUReg\_pd\_gmac\_reset | 0x0 | RW1C | 0x1: Indicates that a gmac power domain reset has happened. |
| [10] | RCUReg\_pd\_sdc\_reset | 0x0 | RW1C | 0x1: Indicates that a sdc power domain reset has happened. |
| [9] | RCUReg\_pd\_sys\_reset | 0x0 | RW1C | 0x1: Indicates that a sys power domain reset has happened. |
| [8] | RCUReg\_cpu\_nsocdbgreset | 0x0 | RW1C | 0x1: Indicates that a cpu nsocdbgreset has happened. |
| [7] | RCUReg\_cpu\_netmreset | 0x0 | RW1C | 0x1: Indicates that a cpu netmreset has happened. |
| [6] | RCUReg\_cpu\_ndbgreset | 0x0 | RW1C | 0x1: Indicates that a cpu ndbgreset has happened. |
| [5] | RCUReg\_cpu\_ncorereset | 0x0 | RW1C | 0x1: Indicates that a cpu ncorereset has happened. |
| [4] | RCUReg\_cpu\_nl2reset | 0x0 | RW1C | 0x1: Indicates that a cpu nl2reset has happened. |
| [3] | RCUReg\_cpu\_ncoreporeset | 0x0 | RW1C | 0x1: Indicates that a cpu ncoreporeset has happened. |
| [2] | RCUReg\_btn\_reset | 0x0 | RW1C | 0x1: Indicates that a button reset has happened. |
| [1] | RCUReg\_wdog\_reset | 0x0 | RW1C | 0x1: Indicates that a system watchdog reset has happened. |
| [0] | RCUReg\_por\_reset | 0x0 | RW1C | 0x1: Indicates that a POR reset (Ipp\_reset\_b) has happened. |

CCR: CMU Control Register

Address: SCU Base + offset (0x00200)

Table 108.CMU Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:2] | - | 0x0 | RO | Reserved |
| [0] | CMUReg\_adv\_clk\_gated | 0x0 | RW | Advanced clock gating mechanism 0x0: always release clock to NOC 0x1: clock to NOC is gated when Bus is idle. (This is an Engineering testing function, don’t release) |

CSR: CMU Status Register

Address: SCU Base + offset (0x00204)

Table 109.CMU Status Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:5] | - | - | RO | Reserved |
| [4] | CMUReg\_clk\_24m\_switch | 0x0 | RO | 0x1: osc clk 24m |
| [3] | CMUReg\_video\_pll\_lock | 0x0 | RO | 0x1: Indicates VIDEO PLL is stable. |
| [2] | CMUReg\_sys\_pll\_lock | 0x0 | RO | 0x1: Indicates SYSTEM PLL is stable. |
| [1] | CMUReg\_ddr\_pll\_lock | 0x0 | RO | 0x1: Indicates DDR PLL is stable. |
| [0] | CMUReg\_cpu\_pll\_lock | 0x0 | RO | 0x1: Indicates CPU PLL is stable. |

CPLL\_CR0: CMU CPU PLL Configuration Register 0

Address: SCU Base + offset (0x00208)

Table 110.CMU CPU PLL Configuration Register 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:27] | - | 0x0 | RO | Reserved |
| [26:24] | cpu\_pll\_POSTDIV2 | 0x1 | RW | PLL post divide 2 setting (1 to 7) Total post divide is POSTDIV1\* POSTDIV2 |
| [23] | - | 0 | RO | Reserved |
| [22:20] | cpu\_pll\_POSTDIV1 | 0x3 | RW | PLL post divide 1 setting (1 to 7) Total post divide is POSTDIV1\* POSTDIV2 |
| [19:8] | cpu\_pll\_FBDIV | 0x64 | RW | PLL Feedback divide value (16 to 2400 in integer mode, 20 to 240 in fractional mode) |
| [7] | - | 0x0 | RO | Reserved |
| [6] | cpu\_pll\_FOUTVCOPD | 0x1 | RW | 0x1: VCO rate output clock power down |
| [5] | cpu\_pll\_FOUTPOSTDIVPD | 0x0 | RW | 0x1: Post divide power down |
| [4] | cpu\_pll\_FOUT4PHASEPD | 0x0 | RW | 0x1: Power down of 4 phase clock generator Note: 4 phase output is also powered down by FOUTPOSTDIVPD |
| [3] | cpu\_pll\_DSMPD | 0x0 | RW | Power down Delta-Sigma Modulator 0x0 : DSM is active. 0x1 : DSM is powered down. |
| [2] | cpu\_pll\_DACPD | 0x0 | RW | Power down noise canceling DAC in FRAC mode 0x0: DAC is active (default mode) 0x1: DAC is not active (test mode only) |
| [1] | cpu\_pll\_BYPASS | 0x0 | RW | 0x1: FREF is bypassed to FOUTPOSTDIV |
| [0] | cpu\_pll\_PD | 0x0 | RW | 0x1: Power Down for PLL |

CPLL\_CR1: CMU CPU PLL Configuration Register 1

Address: SCU Base + offset (0x0020C)

Table 111.CMU CPU PLL Configuration Register 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:30] | - | 0x0 | RO | Reserved |
| [29:24] | cpu\_pll\_REFDIV | 0x1 | RW | Reference divide value (1 to 63) |
| [23:0] | cpu\_pll\_FRAC | 0x0 | RW | Fractional portion of feedback divide value. |

CPLL\_CR2: CMU CPU PLL Configuration Register2

Address: SCU Base + offset (0x00210)

Table 112.CMU CPU PLL Configuration Register2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | RO | Reserved |
| [23] | dssmod\_RESET | 0x0 | RW | Reset modulator state |
| [22] | dssmod\_SEL\_EXTWAVE | 0x0 | RW | Wave Table Select 1’b0 –> Use internal 128 point triangular table (Use EXTWAVEVAL input) 1’b1 –> Use external wave table |
| [21] | dssmod\_DOWNSPREAD | 0x0 | RW | Downspread control 1’b0 –> Center-Spread 1’b1 –> Downspread |
| [20] | dssmod\_DISABLE\_SSCG | 0x0 | RW | SSMOD Bypass 1’b0 –> Use Modulator Output 1’b1 –> Bypass modulator (INTMOD = INTIN, FRACMOD = FRACIN) |
| [19:12] | dssmod\_EXT\_MAXADDR | 0xff | RW | Maximum Address for external wave table |
| [11:9] | - | 0x0 | RO | Reserved |
| [8:4] | dssmoc\_SPREAD | 0x0 | RW | Spread Depth Control in 0.1% steps 5’b00001 –> 0.1% 5’b00010 –> 0.2% ... 5’b10000 –> 1.6% ... 5’b11111 –> 3.1% |
| [3:0] | dssmod\_DIVVAL | 0x5 | RW | Input divider to set modulation frequency |

DPLL\_CR0: CMU DDR PLL Configuration Register 0

Address: SCU Base + offset (0x00214)

Table 113.CMU CPU PLL Configuration Register 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:27] | - | 0x0 | RO | Reserved |
| [26:24] | CMUReg\_ddr\_pll\_POSTDIV2 | 0x1 | RW | PLL post divide 2 setting (1 to 7)  Total post divide is POSTDIV1\* POSTDIV2 |
| [23] | - | 0x0 | RO | Reserved |
| [22:20] | CMUReg\_ddr\_pll\_POSTDIV1 | 0x2 | RW | PLL post divide 1 setting (1 to 7)  Total post divide is POSTDIV1\* POSTDIV2 |
| [19:8] | CMUReg\_ddr\_pll\_FBDIV | 0x37 | RW | PLL Feedback divide value (16 to 2400 in integer mode, 20 to 240 in fractional mode) |
| [7] | - | 0x0 | RO | Reserved |
| [6] | CMUReg\_ddr\_pll\_FOUTVCOPD | 0x1 | RW | 0x1: VCO rate output clock power down |
| [5] | CMUReg\_ddr\_pll\_FOUTPOSTDIVPD | 0x0 | RW | 0x1: Post divide power down |
| [4] | CMUReg\_ddr\_pll\_FOUT4PHASEPD | 0x0 | RW | 0x1: Power down of 4 phase clock generator  Note: 4 phase output is also powered down by FOUTPOSTDIVPD |
| [3] | CMUReg\_ddr\_pll\_DSMPD | 0x1 | RW | Power down Delta-Sigma Modulator  0x0 : DSM is active.  0x1 : DSM is powered down. |
| [2] | CMUReg\_ddr\_pll\_DACPD | 0x0 | RW | Power down noise canceling DAC in FRAC mode  0x0: DAC is active (default mode)  0x1: DAC is not active (test mode only) |
| [1] | CMUReg\_ddr\_pll\_BYPASS | 0x0 | RW | 0x1: FREF is bypassed to FOUTPOSTDIV |
| [0] | CMUReg\_ddr\_pll\_PD | 0x0 | RW | 0x1: Power Down for PLL |

DPLL\_CR1: CMU DDR PLL Configuration Register1

Address: SCU Base + offset (0x00218)

Table 114.CMU DDR PLL Configuration Register1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:30] | - | 0x0 | RO | Reserved |
| [29:24] | CMUReg\_ddr\_pll\_REFDIV | 0x2 | RW | Reference divide value (1 to 63) |
| [23:0] | CMUReg\_ddr\_pll\_FRAC | 0x8E3AB5 | RW | Fractional portion of feedback divide value. |

DPLL\_CR2: CMU DDR PLL Configuration Register2

Address: SCU Base + offset (0x0021C)

Table 115.CMU DDR PLL Configuration Register2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | RO | Reserved |
| [19:12] | dssmod\_EXT\_MAXADDR | 0xff | RW | Maximum Address for external wave table |
| [23] | dssmod\_RESET | 0x0 | RW | Reset modulator state |
| [22] | dssmod\_SEL\_EXTWAVE | 0x0 | RW | Wave Table Select 1’b0 –> Use internal 128 point triangular table (Use EXTWAVEVAL input) 1’b1 –> Use external wave table |
| [21] | dssmod\_DOWNSPREAD | 0x0 | RW | Downspread control 1’b0 –> Center-Spread 1’b1 –> Downspread |
| [20] | dssmod\_DISABLE\_SSCG | 0x0 | RW | SSMOD Bypass 1’b0 –> Use Modulator Output 1’b1 –> Bypass modulator (INTMOD = INTIN, FRACMOD = FRACIN) |
| [11:9] | - | 0x0 | RO | Reserved |
| [8:4] | dssmoc\_SPREAD | 0x0 | RW | Spread Depth Control in 0.1% steps 5’b00001 –> 0.1% 5’b00010 –> 0.2% ... 5’b10000 –> 1.6% ... 5’b11111 –> 3.1% |
| [3:0] | dssmod\_DIVVAL | 0x5 | RW | Input divider to set modulation frequency |

SPLL\_CR0: CMU system PLL Configuration Register0

Address: SCU Base + offset (0x00220)

Table 116.CMU system PLL Configuration Register0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:27] | - | 0x0 | RO | Reserved |
| [26:24] | CMUReg\_sys\_pll\_POSTDIV2 | 0x1 | RW | PLL post divide 2 setting (1 to 7)  Total post divide is POSTDIV1\* POSTDIV2 |
| [23] | - | 0x0 | RO | Reserved |
| [22:20] | CMUReg\_sys\_pll\_POSTDIV1 | 0x3 | RW | PLL post divide 1 setting (1 to 7)  Total post divide is POSTDIV1\* POSTDIV2 |
| [19:8] | CMUReg\_sys\_pll\_FBDIV | 0x53 | RW | PLL Feedback divide value (16 to 2400 in integer mode, 20 to 240 in fractional mode) |
| [7] | - | 0x0 | RO | Reserved |
| [6] | CMUReg\_sys\_pll\_FOUTVCOPD | 0x1 | RW | 0x1: VCO rate output clock power down |
| [5] | CMUReg\_sys\_pll\_FOUTPOSTDIVPD | 0x0 | RW | 0x1: Post divide power down |
| [4] | CMUReg\_sys\_pll\_FOUT4PHASEPD | 0x0 | RW | 0x1: Power down of 4 phase clock generator  Note: 4 phase output is also powered down by FOUTPOSTDIVPD |
| [3] | CMUReg\_sys\_pll\_DSMPD | 0x1 | RW | Power down Delta-Sigma Modulator  0x0 : DSM is active.  0x1 : DSM is powered down. |
| [2] | CMUReg\_sys\_pll\_DACPD | 0x0 | RW | Power down noise canceling DAC in FRAC mode  0x0: DAC is active (default mode)  0x1: DAC is not active (test mode only) |
| [1] | CMUReg\_sys\_pll\_BYPASS | 0x0 | RW | 0x1: FREF is bypassed to FOUTPOSTDIV |
| [0] | CMUReg\_sys\_pll\_PD | 0x0 | RW | 0x1: Power Down for PLL |

SPLL\_CR1: CMU system PLL Configuration Register1

Address: SCU Base + offset (0x00224)

Table 117.CMU system PLL Configuration Register1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:30] | - | 0x0 | RO | Reserved |
| [29:24] | CMUReg\_sys\_pll\_REFDIV | 0x1 | RW | Reference divide value (1 to 63) |
| [23:0] | CMUReg\_sys\_pll\_FRAC | 0x55555 | RW | Fractional portion of feedback divide value. |

SPLL\_CR2: CMU SYS PLL Configuration Register2

Address: SCU Base + offset (0x00228)

Table 118.CMU SYS PLL Configuration Register2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | RO | Reserved |
| [23] | dssmod\_RESET | 0x0 | RW | Reset modulator state |
| [22] | dssmod\_SEL\_EXTWAVE | 0x0 | RW | Wave Table Select 1’b0 –> Use internal 128 point triangular table (Use EXTWAVEVAL input) 1’b1 –> Use external wave table |
| [21] | dssmod\_DOWNSPREAD | 0x0 | RW | Downspread control 1’b0 –> Center-Spread 1’b1 –> Downspread |
| [20] | dssmod\_DISABLE\_SSCG | 0x0 | RW | SSMOD Bypass 1’b0 –> Use Modulator Output 1’b1 –> Bypass modulator (INTMOD = INTIN, FRACMOD = FRACIN) |
| [19:12] | dssmod\_EXT\_MAXADDR | 0xff | RW | Maximum Address for external wave table |
| [11:9] | - | 0x0 | RO | Reserved |
| [8:4] | dssmoc\_SPREAD | 0x0 | RW | Spread Depth Control in 0.1% steps 5’b00001 –> 0.1% 5’b00010 –> 0.2% ... 5’b10000 –> 1.6% ... 5’b11111 –> 3.1% |
| [3:0] | dssmod\_DIVVAL | 0x5 | RW | Input divider to set modulation frequency |

VPLL\_CR0: CMU VIDEO PLL Configuration Register 0

Address: SCU Base + offset (0x0022C)

Table 119.CMU VIDEO PLL Configuration Register 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:27] | - | 0x0 | RO | Reserved |
| [26:24] | CMUReg\_video\_pll\_POSTDIV2 | 0x1 | RW | PLL post divide 2 setting (1 to 7)  Total post divide is POSTDIV1\* POSTDIV2 |
| [23] | - | 0x0 | RO | Reserved |
| [22:20] | CMUReg\_video\_pll\_POSTDIV1 | 0x1 | RW | PLL post divide 1 setting (1 to 7)  Total post divide is POSTDIV1\* POSTDIV2 |
| [19:8] | CMUReg\_video\_pll\_FBDIV | 0x30 | RW | PLL Feedback divide value (16 to 2400 in integer mode, 20 to 240 in fractional mode) |
| [7] | - | 0x0 | RO | Reserved |
| [6] | CMUReg\_video\_pll\_FOUTVCOPD | 0x1 | RW | 0x1: VCO rate output clock power down |
| [5] | CMUReg\_video\_pll\_FOUTPOSTDIVPD | 0x0 | RW | 0x1: Post divide power down |
| [4] | CMUReg\_video\_pll\_FOUT4PHASEPD | 0x0 | RW | 0x1: Power down of 4 phase clock generator  Note: 4 phase output is also powered down by FOUTPOSTDIVPD |
| [3] | CMUReg\_video\_pll\_DSMPD | 0x0 | RW | Power down Delta-Sigma Modulator  0x0 : DSM is active.  0x1 : DSM is powered down. |
| [2] | CMUReg\_video\_pll\_DACPD | 0x0 | RW | Power down noise canceling DAC in FRAC mode  0x0: DAC is active (default mode)  0x1: DAC is not active (test mode only) |
| [1] | CMUReg\_video\_pll\_BYPASS | 0x0 | RW | 0x1: FREF is bypassed to FOUTPOSTDIV |
| [0] | CMUReg\_video\_pll\_PD | 0x0 | RW | 0x1: Power Down for PLL |

VPLL\_CR1: CMU VIDEO PLL Configuration Register 1

Address: SCU Base + offset (0x00230)

Table 120.CMU VIDEO PLL Configuration Register 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [23:0] | CMUReg\_video\_pll\_FRAC | 0x000000 | RW | Fractional portion of feedback divide value. |
| [29:24] | CMUReg\_video\_pll\_REFDIV | 0x1 | RW | Reference divide value (1 to 63) |
| [31:30] | - | 0x0 | RO | Reserved |

VPLL\_CR2: CMU VIDEO PLL Configuration Register 2

Address: SCU Base + offset (0x00234)

Table 121.CMU VIDEO PLL Configuration Register 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | RO | Reserved |
| [23] | dssmod\_RESET | 0x0 | RW | Reset modulator state |
| [22] | dssmod\_SEL\_EXTWAVE | 0x0 | RW | Wave Table Select 1’b0 –> Use internal 128 point triangular table (Use EXTWAVEVAL input) 1’b1 –> Use external wave table |
| [21] | dssmod\_DOWNSPREAD | 0x0 | RW | Downspread control 1’b0 –> Center-Spread 1’b1 –> Downspread |
| [20] | dssmod\_DISABLE\_SSCG | 0x0 | RW | SSMOD Bypass 1’b0 –> Use Modulator Output 1’b1 –> Bypass modulator (INTMOD = INTIN, FRACMOD = FRACIN) |
| [19:12] | dssmod\_EXT\_MAXADDR | 0xff | RW | Maximum Address for external wave table |
| [11:9] | - | 0x0 | RO | Reserved |
| [8:4] | dssmoc\_SPREAD | 0x0 | RW | Spread Depth Control in 0.1% steps 5’b00001 –> 0.1% 5’b00010 –> 0.2% ... 5’b10000 –> 1.6% ... 5’b11111 –> 3.1% |
| [3:0] | dssmod\_DIVVAL | 0x5 | RW | Input divider to set modulation frequency |

CSSR: CMU Clock Source Select Register

Address: SCU Base + offset (0x00238)

Table 122.CMU Clock Source Select Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:25] | - | 0x0 | RO | Reserved |
| [24] | CMUReg\_video\_pll\_sel | 0x0 | RW | Clock source selection of VIDEO PLL out:  0x0: select “osc\_clk”  0x1: select vpll\_FOUTPOSTDIV |
| [23:20] | - | 0x0 | RO | Reserved |
| [19:16] | CMUReg\_sys\_pll\_sel | 0x0 | RW | Clock source selection of SYS PLL out:  0x0: select “osc\_clk”  0x1: select spll\_FOUTPOSTDIV |
| [15:13] | - | 0x0 | RO | Reserved |
| [12] | CMUReg\_ddr\_pll\_sel | 0x0 | RW | Clock source selection of DDR PLL out:  0x0: select “osc\_clk”  0x1: select dpll\_ FOUTPOSTDIV |
| [11:7] | - | 0x0 | RO | Reserved |
| [6:3] | CMUReg\_cpu\_pll\_sel | 0x0 | RW | Clock source selection of CPU PLL out:  0x0: select “osc\_clk”  0x1: select cpll\_FOUTPOSTDIV |
| [2] | - | 0x0 | RO | Reserved |
| [1] | CMUReg\_oscclk\_sel | 0x0 | RW | Osc clk selection, valid when CMUReg\_oscclk\_ctl is set to 0x1  0x0: select osc\_clk24M  0x1: select osc\_clk32K |
| [0] | CMUReg\_oscclk\_ctl | 0x0 | RW | Osc clk source select control  0x0 : control by PMU  0x1: control by CMUReg\_oscclk\_sel |

MSSR: CMU MUX Select Setting Register

Address: SCU Base + offset (0x0023C)

Table 123.CMU MUX Select Setting Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | Reserved |
| [7] | CMUReg\_uart\_clk\_src\_sel | 0x0 |  | uart clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |
| [6] | CMUReg\_gmac\_clksel\_src\_sel | 0x0 | RW | gmac clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |
| [5] | CMUReg\_sdcmclk\_clk\_src\_sel | 0x0 | RW | sdmclk clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |
| [4] | CMUReg\_apb\_clk\_src\_sel | 0x0 | RW | APB clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |
| [3] | CMUReg\_ahb\_clk\_src\_sel | 0x0 | RW | AHB clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |
| [2] | CMUReg\_axi\_clk\_src\_sel | 0x0 | RW | AXI clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |
| [1] | CMUReg\_ddr\_clk\_src\_sel | 0x0 | RW | DDR clock PLL source select control 0x0 : select dpll\_out (default) 0x1: select spll\_out |
| [0] | CMUReg\_cpu\_clk\_src\_sel | 0x0 | RW | CPU clock PLL source select control 0x0 : select cpll\_out (default) 0x1: select spll\_out |

SDSR: CMU SYS Domain Divider Setting Register

Address: SCU Base + offset (0x00240)

Table 124.CMU SYS Domain Divider Setting Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31] | - | 0x0 | RO | Reserved |
| [30:28] | CMUReg\_cpu\_count\_div\_num | 0x3 | RW | cpu count divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [27:26] | CMUReg\_spinor\_noc\_div\_num | 0x1 | RW | spinor noc divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [25:24] | CMUReg\_dma\_noc\_div\_num | 0x1 | RW | spinor divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [23:22] | CMUReg\_isram\_noc\_div\_num | 0x1 | RW | spinor divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [21:20] | CMUReg\_cpu\_noc\_div\_num | 0x0 | RW | spinor divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [19] | - | 0x0 | RO | Reserved |
| [18:16] | CMUReg\_sys\_uart\_div\_num | 0x0 | RW | uart divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [15] | - | 0x0 |  |  |
| [14] | CMUReg\_ddr\_noc\_aclk\_div\_num | 0x0 | RW | ddr noc divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [13:12] | CMUReg\_sys\_ddr\_div\_num | 0x1 | RW | spinor divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [11:10] | CMUReg\_sys\_apb\_div\_num | 0x0 | RW | spinor divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [9:8] | CMUReg\_sys\_spinor\_div\_num | 0x0 | RW | spinor divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [7:6] | CMUReg\_sys\_dma\_div\_num | 0x1 | RW | dma divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [5:4] | CMUReg\_sys\_crypto\_div\_num | 0x1 | RW | crypto divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [3:2] | CMUReg\_sys\_isram\_div\_num | 0x1 | RW | isram divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |
| [1:0] | CMUReg\_sys\_cpu\_div\_num | 0x1 | RW | cpu divider control (sys domain) 0x0 : divided by 1 0x1: divided by 2 0x2: divided by 3,and so on |

DDSR: CMU Debug Domain Divider Setting Register

Address: SCU Base + offset (0x00244)

Table 125.CMU Debug Domain Divider Setting Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | Reserved |
| [7:5] | CMUReg\_cs\_traceclkin\_div\_num | 0x1 | RW | Divider number setting for CoreSight traceclkin 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [4:2] | CMUReg\_cs\_noc \_div\_num | 0x1 | RW | Divider number setting for CoreSight noc 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [1:0] | CMUReg\_cs\_dapclk\_div\_num | 0x1 | RW | Divider number setting for CoreSight dapclk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |

ODSR: CMU Other Domain Divider Setting Register

Address: SCU Base + offset (0x00248)

Table 126.CMU Other Domain Divider Setting Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:30] | - | 0x0 | RO | Reserved |
| [29:26] | CMUReg\_iva\_clk\_vav\_div\_num |  | RW | Divider number setting for sdc sdmclk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [25:24] | CMUReg\_pio\_apb\_div\_num | 0x0 | RW | Divider number setting for sdc sdmclk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [23:22] | CMUReg\_gmac\_ahb\_div\_num | 0x0 | RW | Divider number setting for sdc sdmclk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [21:16] | CMUReg\_gmac\_clksel\_div\_num | 0x1 | RW | Divider number setting for sdc sdmclk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [15] | - | 0x0 | RO | Reserved |
| [14:12] | CMUReg\_sdc\_sdmclk\_div\_num | 0x0 |  | Divider number setting for sdc sdmclk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [11:10] | CMUReg\_sdc\_ahb\_div\_num | 0x0 | RW | Divider number setting for sdc ahb clk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [9:8] | CMUReg\_iva\_clk\_div\_num | 0x3 | RW | Divider number setting for iva clk 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [7] | - | 0x0 | RO | Reserved |
| [6:4] | CMUReg\_iva\_clk\_flt\_div\_num | 0x5 | RW | Divider number setting for iva clk flt 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [3:2] | CMUReg\_iva\_clk\_clf\_2x\_div\_num | 0x1 | RW | Divider number setting for iva clk clf 2x 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |
| [1:0] | CMUReg\_sys\_iva\_div\_num | 0x0 | RW | Divider number setting for iva 0x0: bypass root clock, divide by 1 0x1: divided by 2 0x2: divided by 3, and so on. |

MHIPR: CMU MUX Handshake In-Process Register

Address: SCU Base + offset (0x0024C)

Table 127.CMU MUX Handshake In-Process Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] |  |  |  |  |
| [7] | uart\_clk\_sw\_busy | 0x0 | RO | 0x1: Multiplexer is busy with handshaking process with module. The value read in the multiplexer represents the previous value |
| [6] | sdmclk\_clk\_sw\_busy | 0x0 | RO | 0x0: Multiplexer is switched to new value and stable |
| [5] | gmac\_clksel\_sw\_busy | 0x0 | RO | 0x0: Multiplexer is switched to new value and stable |
| [4] | apb\_clk\_sw\_busy | 0x0 | RO | ‘Busy’ status of ‘sys\_clk\_sw’ multiplexing: |
| [3] | ahb\_clk\_sw\_busy | 0x0 | RO | 0x1: Multiplexer is busy with handshaking process with module. The value read in the multiplexer represents the previous value |
| [2] | axi\_clk\_sw\_busy | 0x0 | RO | 0x0: Multiplexer is switched to new value and stable |
| [1] | ddr\_clk\_sw\_busy | 0x0 | RO | ‘Busy’ status of ‘sys\_clk\_src’ multiplexing: |
| [0] | cpu\_clk\_sw\_busy | 0x0 | RO | Read ‘0’ |

DHIPR: CMU Divider Handshake In-Process Register

Address: SCU Base + offset (0x00250)

Table 128.CMU Divider Handshake In-Process Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:27] |  | 0x0 | RW |  |
| [26] | iva\_clk\_vav\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘iva\_clk\_vav\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x27: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [25] | iva\_clk\_flt\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘iva\_clk\_flt\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x26: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [24] | iva\_clk\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘iva\_clk\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x25: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [23] | iva\_clk\_clf\_2x\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘iva\_clk\_clf\_2x\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x24: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [22] | sys\_iva\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_iva\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x23: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [21] | cpu\_count\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘cpu\_count\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x22: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [20] | ddr\_noc\_aclk\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘ddr\_noc\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x21: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [19] | spinor\_noc\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘spinor\_noc\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x20: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [18] | dma\_noc\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘dma\_noc\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x19: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [17] | isram\_noc\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘isram\_noc\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x18: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [16] | cs\_traceclkin\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘cs\_traceclkin\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x17: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [15] | cs\_noc\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘cs\_noc\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x16: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [14] | cs\_dapclk\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘cs\_dapclk\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x15: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [13] | cpu\_noc\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘cpu\_noc\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x14: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [12] | sdc\_sdmclk\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sdc\_sdmclk\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x13: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [11] | sdc\_ahb\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sdc\_ahb\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x12: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [10] | gmac\_clksel\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘gmac\_clksel\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x11: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [9] | gmac\_ahb\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘gmac\_ahb\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x10: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [8] | pio\_apb\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘pio\_apb\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x9: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [7] | sys\_uart\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_uart\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x8: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [6] | sys\_ddr\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_ddr\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x7: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [5] | sys\_apb\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_apb\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x6: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [4] | sys\_spinor\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_spinor\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x5: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [3] | sys\_dma\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_dma\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x4: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [2] | sys\_crypto\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_crypto\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x3: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [1] | sys\_isram\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_isram\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x2: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |
| [0] | sys\_cpu\_div\_busy | 0x0 | RW | ‘Busy’ status of ‘sys\_cpu\_div\_busy’ mux: 0x0: MUX is switched to new value and stable 0x1: MUX is busy with handshaking process with module. The value read in the MUX represents the previous value |

SYSGR: CMU SYS Domain Clock Gating Register

Address: SCU Base + offset (0x00254)

Table 129.CMU SYS Domain Clock Gating Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:25] | - | - | RO | Reserved |
| [24] | CMUReg\_pmu\_pclk\_en | 0x1 | RW | ‘clk\_pmu\_pclk’ enables |
| [23] | CMUReg\_iomux\_pclk\_en | 0x1 | RW | ‘clk\_iomux\_pclk’ enables |
| [22] | CMUReg\_scu\_pclk\_en | 0x1 | RW | ‘clk\_scu\_pclk’ enables |
| [21] | CMUReg\_gpio\_pclk\_en | 0x1 | RW | ‘clk\_gpio\_pclk’ enables |
| [20] | CMUReg\_uart\_euclk\_en | 0x1 | RW | ‘clk\_uart\_euclk’ enables |
| [19] | CMUReg\_uart\_pclk\_en | 0x1 | RW | ‘clk\_uart\_pclk’ enables |
| [18] | CMUReg\_i2c0\_pclk\_en | 0x1 | RW | ‘clk\_i2c0\_pclk’ enables |
| [17] | CMUReg\_timer\_pclk\_en | 0x1 | RW | ‘clk\_timer\_pclk’ enables |
| [16] | CMUReg\_wdt\_pclk\_en | 0x1 | RW | ‘clk\_wdt\_pclk’ enables |
| [15:13] | - | - | RO | Reserved |
| [12] | CMUReg\_rom\_hclk\_en | 0x1 | RW | ‘clk\_rom\_pclk’ enables |
| [11] | CMUReg\_spinor\_pclk\_en | 0x1 | RW | ‘clk\_spinor\_pclk’ enables |
| [10] | CMUReg\_spinor\_hclk\_en | 0x1 | RW | ‘clk\_spinor\_hclk’ enables |
| [9] | CMUReg\_ddr\_axi\_aclk\_en | 0x1 | RW | ‘clk\_ cs\_ctmclk’ enables |
| [8] | CMUReg\_ddr\_clk\_en | 0x1 | RW | ‘clk\_ cs\_pclkdbg’ enables |
| [7] | - | - | RO |  |
| [6] | CMUReg\_crypto\_aclk\_en | 0x1 | RW | ‘clk\_ cs\_traceclk’ enables |
| [5] | CMUReg\_isram\_mclk1\_en | 0x1 | RW | ‘clk\_isram\_mclk1’ enables |
| [4] | CMUReg\_isram\_mclk0\_en | 0x1 | RW | ‘clk\_isram\_mclk0’ enables |
| [3] | CMUReg\_isram\_aclk\_en | 0x1 | RW | ‘clk\_isram\_aclk’ enables |
| [2] | - | 0x0 | RO | - |
| [1] | CMUReg\_dma\_aclk\_en | 0x1 | RW | ‘clk\_ dma\_aclk’ enables |
| [0] | CMUReg\_cpu\_clkin\_en | 0x1 | RW | ‘clk\_cpu\_clkin’ enables |

DBGGR: CMU DBG Domain Clock Gating Register

Address: SCU Base + offset (0x00258)

Table 130.CMU DBG Domain Clock Gating Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:3] | - | 0x0 | RO | Reserved |
| [2] | CMUReg\_cs\_traceclkin\_en | 0x1 | RW | ‘clk\_ cs\_traceclk’ enables |
| [1] | CMUReg\_cs\_pclks\_en | 0x1 | RW | ‘clk\_ cs\_swclktck’ enables |
| [0] | CMUReg\_cs\_dapclk\_en | 0x1 | RW | ‘clk\_ cs\_dapclk’ enables |

PIOGR: CMU PIO Domain Clock Gating Register

Address: SCU Base + offset (0x0025C)

Table 131.CMU PIO Domain Clock Gating Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:4] | - | 0x0 | RO | Reserved |
| [3] | CMUReg\_pwm\_pclk\_en | 0x1 | RW | ‘clk\_ cs\_pclkdbg’ enables |
| [2] | CMUReg\_spi1\_pclk\_en | 0x1 | RW | ‘clk\_ cs\_traceclk’ enables |
| [1] | CMUReg\_spi0\_pclk\_en | 0x1 | RW | ‘clk\_ cs\_swclktck’ enables |
| [0] | CMUReg\_i2c1\_pclk\_en | 0x1 | RW | clk\_ i2c\_pclk enables |

OGR: CMU Other Domain Clock Gating Register

Address: SCU Base + offset (0x00260)

Table 132.CMU Other Domain Clock Gating Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:14] | - | 0x0 | RO | - |
| [13] | CMUReg\_iva\_clk\_vav\_en | 0x1 | RW | ‘clk\_iva\_clk\_vav’ enable |
| [12] | CMUReg\_iva\_clk\_flt\_en | 0x1 | RW | ‘clk\_iva\_clk\_flt’ enable |
| [11] | CMUReg\_iva\_clk\_en | 0x1 | RW | ‘clk\_iva\_clk’ enable |
| [10] | - | 0x0 | RO | - |
| [9] | CMUReg\_sdc\_sdmclk\_en | 0x0 | RW | ‘clk\_sdc\_sdmclk’ enable |
| [8] | CMUReg\_sdc\_hclk\_en | 0x1 | RW | ‘clk\_sdc\_hclk’ enable |
| [7:6] | - | 0x0 | RO | - |
| [5] | CMUReg\_gmac\_dlyclk\_en | 0x1 | RW | ‘clk\_gmac\_dlyclk’ enables |
| [4] | CMUReg\_gmac\_clksel\_en | 0x1 | RW | ‘clk\_gmac\_clksel’ enables |
| [3] | CMUReg\_gmac\_hclk\_en | 0x1 | RW | ‘clk\_gmac\_hclk’ enables |
| [2] | - | 0x0 | RO | - |
| [1] | CMUReg\_iva\_clk\_clf\_en | 0x1 | RW | ‘clk\_iva\_clk\_clf’ enables |
| [0] | CMUReg\_iva\_clk\_clf\_2x\_en | 0x1 | RW | ‘clk\_iva\_clk\_clf\_2x’ enables |

NGR: CMU NOC Clock Gating Register

Address: SCU Base + offset (0x00264)

Table 133.CMU NOC Clock Gating Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:16] | - | 0x0 | RO | Reserved |
| [15] | CMUReg\_noc\_pioapb\_pclk\_en | 0x1 | RW | ‘clk\_noc\_pioapb\_pclk’ enables |
| [14] | CMUReg\_noc\_sysapb\_pclk\_en | 0x1 | RW | ‘clk\_noc\_sysapb\_pclk’ enables |
| [13] | CMUReg\_noc\_sdc\_hclk\_en | 0x1 | RW | ‘clk\_noc\_sdc\_hclk’ enables |
| [12] | CMUReg\_noc\_gmac\_hclk\_en | 0x1 | RW | ‘clk\_noc\_gmac\_hclk’ enables |
| [11] | CMUReg\_noc\_iva\_pclk\_en | 0x1 | RW | ‘clk\_noc\_iva\_pclk’ enables |
| [10] | CMUReg\_noc\_iva\_aclk\_en | 0x1 | RW | ‘clk\_noc\_iva\_aclk’ enables |
| [9] | CMUReg\_noc\_ddr\_aclk\_en | 0x1 | RW | ‘clk\_noc\_ddr\_aclk’ enables |
| [8] | CMUReg\_noc\_spinor\_pclk\_en | 0x1 | RW | ‘clk\_noc\_spinor\_pclk’ enables |
| [7] | CMUReg\_noc\_sys\_hclk\_en | 0x1 | RW | ‘clk\_noc\_sys\_hclk’ enables |
| [6] | CMUReg\_noc\_crypto\_aclk\_en | 0x1 | RW | ‘clk\_noc\_crypto\_aclk’ enables |
| [5] | CMUReg\_noc\_cs\_pclk\_en | 0x1 | RW | ‘clk\_noc\_isram\_mclk1’ enables |
| [4] | CMUReg\_noc\_isram\_pclk\_en | 0x1 | RW | ‘clk\_noc\_isram\_mclk0’ enables |
| [3] | CMUReg\_noc\_isram\_aclk\_en | 0x1 | RW | ‘clk\_noc\_isram\_aclk’ enables |
| [2] | CMUReg\_noc\_dma\_pclk\_en | 0x1 | RW | ‘clk\_noc\_dma\_pclk’ enables |
| [1] | CMUReg\_noc\_dma\_aclk\_en | 0x1 | RW | ‘clk\_noc\_dma\_aclk’ enables |
| [0] | CMUReg\_noc\_cpu\_clkin\_en | 0x1 | RW | ‘clk\_noc\_cpu\_clkin’ enables |

ICR0: ISRAM Control Register0

Address: SCU Base + offset (0x00304)

Table 134.ISRAM Control Register0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:27] | - | 0x0 | RW | Reserved |
| [26] | ISRAM\_cactive | 0x0 | RO | AXI low-power interface signals |
| [25] | ISRAM\_csysack | 0x0 | RO | AXI low-power interface signals |
| [24] | ISRAM\_csysreq | 0x1 | RW | AXI low-power interface signals |
| [23:16] | ISRAM\_user\_config | 0x0 | RO | General purpose output signals that you program using the User Config Register on page 3-24 |
| [15:14] | - | 0x0 | RW | Reserved |
| [13:12] | ISRAM\_sram\_mw\_1 | 0x2 | RW | Sets the memory width for chip select 0, on memory interface <x>, when remap\_<x> is HIGH. The encoding is: b00 = 8-bit b01 = 16-bit b10 = 32-bit b11 = Reserved. |
| [11] | ISRAM\_mux\_mode\_1 | 0x0 | RW | When HIGH, the memory interface operates in multiplexed address/data mode. |
| [10] | ISRAM\_remap\_1 | 0x0 | RW | When HIGH, the SMC remaps chip select 0, on memory interface <x>, to address 0x0. |
| [9:8] | ISRAM\_sram\_mw\_0 | 0x2 | RW | Sets the memory width for chip select 0, on memory interface <x>, when remap\_<x> is HIGH. The encoding is: b00 = 8-bit b01 = 16-bit b10 = 32-bit b11 = Reserved. |
| [7] | ISRAM\_mux\_mode\_0 | 0x0 | RW | When HIGH, the memory interface operates in  multiplexed address/data mode. |
| [6] | ISRAM\_remap\_0 | 0x0 | RW | When HIGH, the SMC remaps chip select 0, on memory interface <x>, to address 0x0. |
| [5] | ISRAM\_a\_gt\_m1\_sync | 0x0 | RW | Set this signal HIGH if aclk is greater than mclk0 but is stillsynchronous. |
| [4] | ISRAM\_async1 | 0x1 | RW | When HIGH, indicates to aclk domain that aclk is synchronous to mclk0. When LOW, synchronizing logic is enabled. |
| [3] | ISRAM\_msync1 | 0x1 | RW | Set this signal HIGH if aclk is greater than mclk0 but is still synchronous. |
| [2] | ISRAM\_a\_gt\_m0\_sync | 0x0 | RW | When HIGH, indicates to aclk domain that aclk is synchronous to mclk0. When LOW, synchronizing logic is enabled. |
| [1] | ISRAM\_async0 | 0x1 | RW | When HIGH, indicates to aclk domain that aclk is synchronous to mclk0. When LOW, synchronizing logic is enabled. |
| [0] | ISRAM\_msync0 | 0x1 | RW | When HIGH, indicates to mclk0 domain that mclk0 is synchronous to aclk. When LOW, synchronizing logic is enabled. |

ICR1: ISRAM Control Register1

Address: SCU Base + offset (0x00308)

Table 135.ISRAM Control Register1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit | Name | Reset | Type | Description |
| [31:24] | ISRAM\_address\_match1\_0 | 0x1 | RW | The comparison value that determines the chip select  base address. |
| [23:16] | ISRAM\_address\_mask1\_0 | 0xff | RW | A mask applied to the AXI address bits [31:24] before the comparison with the address\_match value. |
| [15:8] | ISRAM\_address\_match0\_0 | 0x0 | RW | The comparison value that determines the chip select base address. |
| [7:0] | ISRAM\_address\_mask0\_0 | 0xff | RW | A mask applied to the AXI address bits [31:24] before the comparison with the address\_match value. |

DCR0: DMAC Control Register0

Address: SCU Base + offset (0x00310)

Table 136.DMAC Control Register0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:0] | SCU\_boot\_addr | 0x0 | RW | Configures the address location that contains the first instruction the DMAC executes, when it exits from reset The DMAC only uses this address when boot\_from\_pc is HIGH. |

DCR1: DMAC Control Register1

Address: SCU Base + offset (0x00314)

Table 137.DMAC Control Register1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:1] | - | - | RO | Reserved |
| [0] | SCU\_boot\_from\_pc | 0x0 | RW | 0 = DMAC waits for an instruction from either APB interface 1 = DMAC manager thread executes the instruction that is located at the address that boot\_addr[31:0] provides. |

DDRCR: DDR Control Register

Address: SCU Base + offset (0x00318)

Table 138.DDR Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit | Name | Reset | Type | Description |
| [31:28] | - | 0x0 | RO | Reserved |
| [27:22] | lp\_ext\_state | 0x00 | RO | Defines the current low power state, regardless of the method of entry of that state. |
| [21:20] | lp\_ext\_resp | 0x0 | RO | Reserved |
| [19] | lp\_ext\_req | 0x0 | RW | External request to manage the LPC module. |
| [18] | lp\_ext\_priority\_req | 0x0 | RW | High-priority external request to manage the LPC module. |
| [17] | lp\_ext\_done | 0x0 | RO | LPC module indication that the last command from the external pin interface has been completed. |
| [16] | lp\_ext\_cmd\_strb | 0x0 | RW | Once the external request is received and acknowledged, this signal is used to indicate a valid request on the lp\_ext\_cmd signal. |
| [15:8] | lp\_ext\_cmd | 0x00 | RW | Defines the low power command requested through the external pin interface. |
| [7:5] | - | 0x0 | RO | Reserved |
| [4:1] | lp\_ext\_arb\_state | 0x0 | RO | Defines the current manager of the low power module and the status of the lock. |
| [0] | lp\_ext\_ack | 0x0 | RO | LPC module acknowledge of the external request. |

DDRCSR: DDR Control/Status Register

Address: SCU Base + offset (0x0031C)

Table 139.DDR Control/Status Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | Reserved |
| [7] | controller\_busy | 0x0 | RO | Indicator that the controller is processing a command. |
| [6] | refresh\_in\_process | 0x0 | RO | 18 |
| [5] | q\_almost\_full | 0x0 | RO | Indicates that the command queue has reached the value set in the q\_fullness parameter. |
| [4:1] | port\_busy | 0x0 | RO | This signal contains 1 bit per port. A port’s port\_busy bit will only be low when the controller is not reading data, writing data or processing a command for that port. |
| [0] | mem\_rst\_valid | 0x0 | RO | When memory is in self-refresh, this signal is used to indicate that a full memory initialization is not required. It also indicates that the system is driving the memory reset and cke signals. |

CRPTCR: Crypto Control Register

Address: SCU Base + offset (0x00320)

Table 140.Crypto Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:1] | - | 0x0 | RO | Reserved |
| [0] | I\_trng\_osc\_stop\_n | 0x0 | RW | Drive with 1'b0 to stop rings. Drive with 1'b1 to allow rings to oscillate. |

GCR: GMAC Control Register

Address: SCU Base + offset (0x00324)

Table 141.GMAC Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:30] | - | 0x0 | RO | Reserved |
| [29:8] | GMAC\_DELAY\_LINE | 22'h801 | RW | delay line adjustment : 0~8ns |
| [7:6] | - | 0x0 | RO | Reserved |
| [5:2] | GMAC\_INTERFACE\_SEL | 4'b1100 | RW | decide the source of "clksel" and " clk50" of internal  model "mac\_chip" |
| [1:0] | GMAC\_IFCSEL | 2'b01 | RW | phy interface selection,default is RMII |

SPCR: SPINOR Control Register

Address: SCU Base + offset (0x00328)

Table 142.SPINOR Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:14] | - | 0x0 | RO | Reserved |
| [13:8] | SPINOR\_spi\_divr\_ahb | 6'h01 | RW | SPI master controller clock divisor pins, the value of sclk\_divr\_ahb is used to generate the transmit and receive bit rate of the SPI master controller.  The frequency divisor equation for generation sclk is below: sclk divisor = (sclk\_divr\_ahb[5:3] + 1) x 2(sclk\_divr\_ahb[2:0] + 1) sclk = hclk/sclk divisor |
| [7:4] | - | 0x0 | RO | Reserved |
| [3] | SPINOR\_addr\_en4b | 1'b0 | RW | 1’b0: keep in 3 byte address mode  1’b1: enter 4 bytes address mode command(instruction :b7h) , only for Macro nix |
| [2] | SPINOR\_addr\_fmt | 1'b0 | RW | 1’b0: 3 bytes address mode  1’b1: direct 4 bytes address mode |
| [1:0] | SPINOR\_io\_width\_sel | 2'b01 | RW | Select 1/2/4 I/O 2’b00: 1 I/O 2’b01: 2 I/O 2’b10: 4 I/O 2’b11: reserved |

SDRCSR: SDIO Control/status Register

Address: SCU Base + offset (0x0032C)

Table 143.SDIO Control/status Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit | Name | Reset | Type | Description |
| [31:21] | - | 0x0 | RO | Reserved |
| [20] | sdrtr\_s2 | 1'b0 | RW | uhs-I re-tune request |
| [19:16] | sdsm\_s2 | 4'h0 | RO | 18'h0 |
| [15] | sdrtr\_s1 | 1'b0 | RW | uhs-I re-tune request |
| [14:11] | sdsm\_s1 | 4'h0 | RO | speed mode |
| [10] | sdrtr\_s0 | 1'b0 | RW | uhs-I re-tune request |
| [9:6] | sdsm\_s0 | 4'h0 | RO | speed mode |
| [5] | ci\_s2 | 1'b0 | RO | card inserted |
| [4] | css\_s2 | 1'b0 | RO | card state stable |
| [3] | ci\_s1 | 1'b0 | RO | card inserted |
| [2] | css\_s1 | 1'b0 | RO | card state stable |
| [1] | ci\_s0 | 1'b0 | RO | card inserted |
| [0] | css\_s0 | 1'b0 | RO | card state stable |

CAOR0:CA7 Options register0

Address: SCU Base + offset (0x00330)

Table 144.CA7 Options register0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit | Name | Reset | Type | Description |
| [31:16] | - | 0x0 | RO | Reserved |
| [15] | CA7\_DFTRAMHOLD | 1'b0 | RW | - |
| [14] | CA7\_DFTRSTDISABLE | 1'b0 | RW | - |
| [13] | CA7\_DFTSE | 1'b0 | RW | - |
| [12] | SCU\_CA7\_VINITHI | 1'b0 | RW | - |
| [11] | CA7\_SCU\_DBGRESTARTED | 1’b0 | RO | external debug restart ack |
| [10] | SCU\_CA7\_DBGRESTART | 1'b0 | RW | external debug restart |
| [9] | CA7\_SCU\_COMMTX | 1'b0 | RO | external debug |
| [8] | CA7\_SCU\_COMMRX | 1'b0 | RO | external debug |
| [7] | CA7\_SCU\_DBGTRIGGER | 1'b0 | RO | external debug trigger |
| [6] | SCU\_CA7\_EDBGRQ, | 1'b0 | RW | external debug request |
| [5] | SPNIDEN | 1'b0 | RW | Secure privileged non-invasive debug enable: 0 Not enabled. 1 Enabled. |
| [4] | SPIDEN | 1'b0 | RW | Secure privileged invasive debug enable: 0 Not enabled. 1 Enabled. |
| [3] | NIDEN | 1'b0 | RW | Non-invasive debug enable: 0 Not enabled. 1 Enabled. |
| [2] | DBGEN | 1'b0 | RW | Invasive debug enable: 0 Not enabled. 1 Enabled. |
| [1] | CFGSDISABLE | 1'b0 | RW | Disables write access to some secure GIC registers. 0 = not enabled 1 = enabled. |
| [0] | CFGTE | 1'b0 | RW | Default exception handling state. It sets the initial value of the TE bit in the CP15 System Control Register (SCTLR): 0 TE bit is LOW. 1 TE bit is HIGH. This pin is only sampled during reset of the processor. |

CAOR1:CA7 Options register1

Address: SCU Base + offset (0x00334)

Table 145.CA7 Options register1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:0] | CNTVALUEB[63:32] | 32'h0 | RW | Global  system  counter  value  in binary  format(AON domain, clock domain crossing) |

CAOR2:CA7 Options register2

Address: SCU Base + offset (0x00338)

Table 146.CA7 Options register2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit | Name | Reset | Type | Description |
| [31:0] | CNTVALUEB[31:0] | 32'h0 | RW | Global  system  counter  value  in binary  format(AON domain, clock domain crossing) |

CTCR3: CoreSight Trace Control Register3

Address: SCU Base + offset (0x00348)

Table 147.CA7 Options register3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:12] | - | 0x0 | RO | Reserved |
| [11] | CA7\_status\_jtagntop | 0x0 | RO | JTAG state machine is in one of the four modes:  Test-Logic-Reset.  Run-Test/Idle.  Select-DR-Scan  Select-IR-Scan Connect SCU reg Type R |
| [10] | CA7\_status\_jtagnsw | 0x0 | RO | HIGH If JTAG selected. LOW If SWD selected. Connect SCU reg Type R |
| [9] | SCU\_csysack | 0x0 | RO | Power down ack from apb async bridge |
| [8] | SCU\_csysreq | 0x1 | RW | Power down request to apb async bridge |
| [7:3] | CA7\_configuration\_staticcfg\_tpmaxdatasize | 0xF | RW | Configure trace port data size |
| [2] | SCU\_CS\_ipt\_rst\_bypass | 0x0 | RW | H: bypass rst signal |
| [1] | SCU\_niden | 0x0 | RW | Non-invasive debug enable Connect SCU reg Type Rw |
| [0] | SCU\_dbgen | 0x0 | RW | Invasive debug enable. Connect SCU reg Type Rw |

CTCR1: CoreSight Control Register

Address: SCU Base + offset (0x0034C)

Table 148.CoreSight Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:0] | SCU\_staticcfg\_rombaseaddr | 32'h8000\_0003 | RW | Rombaseaddr for APB-AP |

CTCR2: CoreSight Control Register

Address: SCU Base + offset (0x00350)

Table 149.CoreSight Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:0] | SCU\_staticcfg\_targetid | 32'h088C\_001D | RW | TargetId for CS |

SPRCR: SPINOR Reset Control Register

Address: SCU Base + offset (0x00358)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:28] | - | 0x0 | RO | Reserved |
| [27:0] | rst\_wait\_cnt | 28'd93750 (28'h16E36) | RW | spinor reset wait count |

SDCR1:SDIO control register

Address: SCU Base + offset (0x0035C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:29] | - | 0x0 | RO | Reserved |
| [28] | REG\_SDIO\_sdmclk\_value1 | 0x0 | RW | Slot1 tuning sample result output |
| [27:26] | REG\_SDIO\_selphase1 | 0x0 | RW | Slot1 phase select between two step  1/2/3/4 phase |
| [25:24] | REG\_SDIO\_selmode1 | 0x0 | RW | Slot1 mode select for software  sss/fff/normal/test mode |
| [23:16] | REG\_SDIO\_stepstart1 | 0x0 | RW | Slot1 clk tuning start phase |
| [15:13] | - | 0x0 | RO | Reserved |
| [12] | REG\_SDIO\_sdmclk\_value0 | 0x0 | RW | Slot0 tuning sample result output |
| [11:10] | REG\_SDIO\_selphase0 | 0x0 | RW | Slot0 phase select between two step  1/2/3/4 phase |
| [9:8] | REG\_SDIO\_selmode0 | 0x0 | RW | Slot0 mode select for software  sss/fff/normal/test mode |
| [7:0] | REG\_SDIO\_stepstart0 | 0x0 | RW | Slot0 clk tuning start phase |

1. Power Management Unit
   1. Register Summary

Table 150.PMU Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| GCR0 | 0x00 | 0x00000000 | Global control register0 |
| GCR1 | 0x04 | 0x00000000 | Global control register1 |
| SDC\_PUPSCR | 0x20 | 0x000fff0f | SDC domain power up sequence control register |
| SDC\_PDNSCR | 0x24 | 0x000fff0f | SDC domain power down sequence control register |
| SDC\_NOCSCR | 0x28 | 0x000000ff | SDC domain NIU sequence control register |
| SDC\_PSR | 0x2C | 0x00850000 | SDC domain power status register |
| GMAC\_PUPSCR | 0x30 | 0x000fff0f | GMAC domain power up sequence control register |
| GMAC\_PDNSCR | 0x34 | 0x000fff0f | GMAC domain power down sequence control register |
| GMAC\_NOCSCR | 0x38 | 0x000000ff | GMAC domain NIU sequence control register |
| GMAC\_PSR | 0x3C | 0x00850000 | GMAC domain power status register |
| PIO\_PUPSCR | 0x40 | 0x000fff0f | PIO domain power up sequence control register |
| PIO\_PDNSCR | 0x44 | 0x000fff0f | PIO domain power down sequence control register |
| PIO\_NOCSCR | 0x48 | 0x000000ff | PIO domain NIU sequence control register |
| PIO\_PSR | 0x4C | 0x00850000 | PIO domain power status register |
| DBG\_PUPSCR | 0x50 | 0x000fff0f | DBG domain power up sequence control register |
| DBG\_PDNSCR | 0x54 | 0x000fff0f | DBG domain power down sequence control register |
| DBG\_NOCSCR | 0x58 | 0x000000ff | DBG domain NIU sequence control register |
| DBG\_PSR | 0x5C | 0x00330000 | DBG domain power status register |
| CPU\_PCR | 0x60 | 0x00000000 | SYS domain power control register |
| CPU\_PDNSCR | 0x64 | 0x000fff0f | SYS domain power up sequence control register |
| CPU\_PDNSCR | 0x68 | 0x000fff0f | SYS domain power down sequence control register |
| CPU\_NOCSCR | 0x6C | 0x000000ff | SYS domain NIU sequence control register |
| CPU\_PSR | 0x70 | 0x00850000 | SYS domain power status register |

* 1. Register Descriptions

GCR0: Global control register0

Address: PMU Base + offset (0x00)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31] | reg\_mask\_gpc\_irq | 0x0 | RW | 0x1: Mask IRQ of GPC/INTC. Set this bit to 1, GPC won’t set IRQ to CPU for all GPC events.  **Note**: GPC/INTC triggers IRQ, not FIQ. |
| [30:26] | - | - | - | reserved |
| [25] | reg\_gpc\_dbg\_cup\_req | 0x0 | RW | DBG clock up request, self clear |
| [24] | reg\_gpc\_dbg\_cdn\_req | 0x0 | RW | DBG clock down request, self clear |
| [23] | reg\_gpc\_pio\_cup\_req | 0x0 | RW | PIO clock up request, self clear |
| [22] | reg\_gpc\_pio\_cdn\_req | 0x0 | RW | PIO clock down request, self clear |
| [21] | reg\_gpc\_gmac\_cup\_req | 0x0 | RW | GMAC clock up request, self clear |
| [20] | reg\_gpc\_gmac\_cdn\_req | 0x0 | RW | GMAC clock down request, self clear |
| [19] | reg\_gpc\_sdc\_cup\_req | 0x0 | RW | SDC clock up request, self clear |
| [18] | reg\_gpc\_sdc\_cdn\_req | 0x0 | RW | SDC clock down request, self clear |
| [17:13] | - | - | - | reserved |
| [12] | reg\_ckih\_pcr | 0x0 | RW | CKIH disable when CPU power down |
| [11:10] | - | - | - | reserved |
| [9] | reg\_gpc\_dbg\_pup\_req | 0x0 | RW | DBG power up request, self clear |
| [8] | reg\_gpc\_dbg\_pdn\_req | 0x0 | RW | DBG power down request, self clear |
| [7] | reg\_gpc\_pio\_pup\_req | 0x0 | RW | PIO power up request, self clear |
| [6] | reg\_gpc\_pio\_pdn\_req | 0x0 | RW | PIO power down request, self clear |
| [5] | reg\_gpc\_gmac\_pup\_req | 0x0 | RW | GMAC power up request, self clear |
| [4] | reg\_gpc\_gmac\_pdn\_req | 0x0 | RW | GMAC power down request, self clear |
| [3] | reg\_gpc\_sdc\_pup\_req | 0x0 | RW | SDC power up request, self clear |
| [2] | reg\_gpc\_sdc\_pdn\_req | 0x0 | RW | SDC power down request, self clear |
| [1:0] | - | - | - | reserved |

GCR1: Global control register1

Address: PMU Base + offset (0x04)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:30] | - | - | - | reserved |
| [29] | reg\_dbg\_ignore\_ack | 0x0 | RW | Ignoring ack from dbg domain power gate |
| [28] | reg\_pio\_ignore\_ack | 0x0 | RW | Ignoring ack from pio domain power gate |
| [27] | reg\_gmac\_ignore\_ack | 0x0 | RW | Ignoring ack from gmac domain power gate |
| [26] | reg\_sdc\_ignore\_ack | 0x0 | RW | Ignoring ack from sdc domain power gate |
| [25] | - | - | - | reserved |
| [24] | reg\_cpu\_ignore\_ack | 0x0 | RW | Ignoring ack from sys domain power gate |
| [23:18] | - | - | - | reserved |
| [17] | reg\_test\_on | 0x0 | RW | Reserved bit for test |
| [16] | reg\_ignore\_irq | 0x0 | RW | Ignoring external irq |
| [15:0] | reg\_slow\_cnt | 0x0 | RW | Divider for GPC internal counter |

SDC\_PUPSCR: SDC domain power up sequence control register

Address: PMU Base + offset (0x20)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_sdc\_sw2iso | 0xfff | RW | These bits define clk24m cycle counts from ‘power switch-ON’ to ‘isolation removal’.  **Note**: SW2ISO must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | reserved |
| [5:0] | reg\_sdc\_up\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-up request assertion (pup\_req)’ to ‘power up sequence’. |

SDC\_PDNSCR: SDC domain power down sequence control register

Address: PMU Base + offset (0x24)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_sdc\_iso2sw | 0xfff | RW | These bits define clk24m cycle counts from ‘isolation assertion’ to ‘power switch-OFF’.  **Note**: ISO2SW must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | Reserved |
| [5:0] | reg\_sdc\_dn\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-down request assertion (pdn\_req)’ to ‘power down sequence’. |

SDC\_NOCSCR: SDC domain NIU sequence control register

Address: PMU Base + offset (0x28)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | reserved |
| [7:0] | reg\_sdc\_noc\_timeout | 0xff | RW | These bits define clk24m cycle counts from ‘noc idle request assertion’ to ‘noc idle timeout’, if noc idle timeout, gpc.pgc\_cpu0 enter poff directly |

SDC\_PSR: SDC domain power status register

Address: PMU Base + offset (0x2C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | - | reserved |
| [23:20] | reg\_gpc\_sdc\_sts | 0x8 | RO | The status bit indicates that sdc performance status |
| 0x0000 : PUP |
| 0x0001 : CUP |
| 0x0010 : WAITACTIVE |
| 0x0011 : RUN |
| 0x0100 : DNREQ |
| 0x0101 : WAITIDLE |
| 0x0110 : COFF |
| 0x0111 : PDN |
| 0x1000 : POFF |
| 0x1001 : OSCDN |
| 0x1010 : OSCOFF |
| 0x1011 : UPREQ |
| 0x1100 : OSCUP |
| 0x1101 : OSCACTIVE |
| [19:16] | reg\_pgc\_sdc\_sts | 0x5 | RO | The status bit indicates that sdc power status |
| Note : this is a status bit. |
| 0x0000 : initial |
| 0x0001 : initial prun |
| 0x0010 : prun |
| 0x0011 : run |
| 0x0100 : poff |
| 0x0101 : off |
| [15:7] | - | 0x0 | - | reserved |
| [6] | reg\_en\_sdc\_timeout | 0x0 | RW | reg\_int\_sdc\_timeout interrupt enable, assert gpc\_int\_n if reg\_int\_sdc\_timeout is 1 |
| [5] | reg\_en\_sdc\_pup | 0x0 | RW | reg\_int\_sdc\_pup interrupt enable, assert gpc\_int\_n if reg\_int\_sdc\_pup is 1 |
| [4] | reg\_en\_sdc\_pdn | 0x0 | RW | reg\_int\_sdc\_pdn interrupt enable, assert gpc\_int\_n if reg\_int\_sdc\_pdn is 1 |
| [3] | - |  |  | reserved |
| [2] | reg\_int\_sdc\_timeout | 0x0 | RW1C | NoC idle sequence timeout status |
| [1] | reg\_int\_sdc\_pup | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power up sequence has done  0x1 : domain was powered up for the previous request |
| [0] | reg\_int\_sdc\_pdn | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power down sequence has done  0x1 : domain was powered down for the previous request |

GMAC\_PUPSCR: GMAC domain power up sequence control register

Address: PMU Base + offset (0x30)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_gmac\_sw2iso | 0xfff | RW | These bits define clk24m cycle counts from ‘power switch-ON’ to ‘isolation removal’.  **Note**: SW2ISO must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | reserved |
| [5:0] | reg\_gmac\_up\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-up request assertion (pup\_req)’ to ‘power up sequence’. |

GMAC\_PDNSCR: GMAC domain power down sequence control register

Address: PMU Base + offset (0x34)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_gmac\_iso2sw | 0xfff | RW | These bits define clk24m cycle counts from ‘isolation assertion’ to ‘power switch-OFF’.  **Note**: ISO2SW must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | Reserved |
| [5:0] | reg\_gmac\_dn\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-down request assertion (pdn\_req)’ to ‘power down sequence’. |

GMAC\_NOCSCR: GMAC domain NIU sequence control register

Address: PMU Base + offset (0x38)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | reserved |
| [7:0] | reg\_gmac\_noc\_timeout | 0xff | RW | These bits define clk24m cycle counts from ‘noc idle request assertion’ to ‘noc idle timeout’, if noc idle timeout, gpc.pgc\_cpu0 enter poff directly |

GMAC\_PSR: GMAC domain power status register

Address: PMU Base + offset (0x3C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | - | reserved |
| [23:20] | reg\_gpc\_gmac\_sts | 0x8 | RO | The status bit indicates that gmac performance status |
| 0x0000 : PUP |
| 0x0001 : CUP |
| 0x0010 : WAITACTIVE |
| 0x0011 : RUN |
| 0x0100 : DNREQ |
| 0x0101 : WAITIDLE |
| 0x0110 : COFF |
| 0x0111 : PDN |
| 0x1000 : POFF |
| 0x1001 : OSCDN |
| 0x1010 : OSCOFF |
| 0x1011 : UPREQ |
| 0x1100 : OSCUP |
| 0x1101 : OSCACTIVE |
| [19:16] | reg\_pgc\_gmac\_sts | 0x5 | RO | The status bit indicates that gmac power status |
| Note : this is a status bit. |
| 0x0000 : initial |
| 0x0001 : initial prun |
| 0x0010 : prun |
| 0x0011 : run |
| 0x0100 : poff |
| 0x0101 : off |
| [15:7] | - | 0x0 | - | reserved |
| [6] | reg\_en\_gmac\_timeout | 0x0 | RW | reg\_int\_gmac\_timeout interrupt enable, assert gpc\_int\_n if reg\_int\_gmac\_timeout is 1 |
| [5] | reg\_en\_gmac\_pup | 0x0 | RW | reg\_int\_gmac\_pup interrupt enable, assert gpc\_int\_n if reg\_int\_gmac\_pup is 1 |
| [4] | reg\_en\_gmac\_pdn | 0x0 | RW | reg\_int\_gmac\_pdn interrupt enable, assert gpc\_int\_n if reg\_int\_gmac\_pdn is 1 |
| [3] | - |  |  | reserved |
| [2] | reg\_int\_gmac\_timeout | 0x0 | RW1C | NoC idle sequence timeout status |
| [1] | reg\_int\_gmac\_pup | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power up sequence has done  0x1 : domain was powered up for the previous request |
| [0] | reg\_int\_gmac\_pdn | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power down sequence has done  0x1 : domain was powered down for the previous request |

PIO\_PUPSCR: PIO domain power up sequence control register

Address: PMU Base + offset (0x40)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_pio\_sw2iso | 0xfff | RW | These bits define clk24m cycle counts from ‘power switch-ON’ to ‘isolation removal’.  **Note**: SW2ISO must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | reserved |
| [5:0] | reg\_pio\_up\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-up request assertion (pup\_req)’ to ‘power up sequence’. |

PIO\_PDNSCR: PIO domain power down sequence control register

Address: PMU Base + offset (0x44)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_pio\_iso2sw | 0xfff | RW | These bits define clk24m cycle counts from ‘isolation assertion’ to ‘power switch-OFF’.  **Note**: ISO2SW must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | Reserved |
| [5:0] | reg\_pio\_dn\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-down request assertion (pdn\_req)’ to ‘power down sequence’. |

PIO\_NOCSCR: PIO domain NIU sequence control register

Address: PMU Base + offset (0x48)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | reserved |
| [7:0] | reg\_pio\_noc\_timeout | 0xff | RW | These bits define clk24m cycle counts from ‘noc idle request assertion’ to ‘noc idle timeout’, if noc idle timeout, gpc.pgc\_cpu0 enter poff directly |

PIO\_PSR: PIO domain power status register

Address: PMU Base + offset (0x4C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | - | reserved |
| [23:20] | reg\_gpc\_pio\_sts | 0x8 | RO | The status bit indicates that pio performance status |
| 0x0000 : PUP |
| 0x0001 : CUP |
| 0x0010 : WAITACTIVE |
| 0x0011 : RUN |
| 0x0100 : DNREQ |
| 0x0101 : WAITIDLE |
| 0x0110 : COFF |
| 0x0111 : PDN |
| 0x1000 : POFF |
| 0x1001 : OSCDN |
| 0x1010 : OSCOFF |
| 0x1011 : UPREQ |
| 0x1100 : OSCUP |
| 0x1101 : OSCACTIVE |
| [19:16] | reg\_pgc\_pio\_sts | 0x5 | RO | The status bit indicates that pio power status |
| Note : this is a status bit. |
| 0x0000 : initial |
| 0x0001 : initial prun |
| 0x0010 : prun |
| 0x0011 : run |
| 0x0100 : poff |
| 0x0101 : off |
| [15:7] | - | 0x0 | - | reserved |
| [6] | reg\_en\_pio\_timeout | 0x0 | RW | reg\_int\_pio\_timeout interrupt enable, assert gpc\_int\_n if reg\_int\_pio\_timeout is 1 |
| [5] | reg\_en\_pio\_pup | 0x0 | RW | reg\_int\_pio\_pup interrupt enable, assert gpc\_int\_n if reg\_int\_pio\_pup is 1 |
| [4] | reg\_en\_pio\_pdn | 0x0 | RW | reg\_int\_pio\_pdn interrupt enable, assert gpc\_int\_n if reg\_int\_pio\_pdn is 1 |
| [3] | - |  |  | reserved |
| [2] | reg\_int\_pio\_timeout | 0x0 | RW1C | NoC idle sequence timeout status |
| [1] | reg\_int\_pio\_pup | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power up sequence has done  0x1 : domain was powered up for the previous request |
| [0] | reg\_int\_pio\_pdn | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power down sequence has done  0x1 : domain was powered down for the previous request |

DBG\_PUPSCR: DBG domain power up sequence control register

Address: PMU Base + offset (0x50)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_dbg\_sw2iso | 0xfff | RW | These bits define clk24m cycle counts from ‘power switch-ON’ to ‘isolation removal’.  **Note**: SW2ISO must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | reserved |
| [5:0] | reg\_dbg\_up\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-up request assertion (pup\_req)’ to ‘power up sequence’. |

DBG\_PDNSCR: DBG domain power down sequence control register

Address: PMU Base + offset (0x54)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_dbg\_iso2sw | 0xfff | RW | These bits define clk24m cycle counts from ‘isolation assertion’ to ‘power switch-OFF’.  **Note**: ISO2SW must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | Reserved |
| [5:0] | reg\_dbg\_dn\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-down request assertion (pdn\_req)’ to ‘power down sequence’. |

DBG\_NOCSCR: DBG domain NIU sequence control register

Address: PMU Base + offset (0x58)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | reserved |
| [7:0] | reg\_dbg\_noc\_timeout | 0xff | RW | These bits define clk24m cycle counts from ‘noc idle request assertion’ to ‘noc idle timeout’, if noc idle timeout, gpc.pgc\_cpu0 enter poff directly |

DBG\_PSR: DBG domain power status register

Address: PMU Base + offset (0x5C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | - | reserved |
| [23:20] | reg\_gpc\_dbg\_sts | 0x8 | RO | The status bit indicates that dbg performance status |
| 0x0000 : PUP |
| 0x0001 : CUP |
| 0x0010 : WAITACTIVE |
| 0x0011 : RUN |
| 0x0100 : DNREQ |
| 0x0101 : WAITIDLE |
| 0x0110 : COFF |
| 0x0111 : PDN |
| 0x1000 : POFF |
| 0x1001 : OSCDN |
| 0x1010 : OSCOFF |
| 0x1011 : UPREQ |
| 0x1100 : OSCUP |
| 0x1101 : OSCACTIVE |
| [19:16] | reg\_pgc\_dbg\_sts | 0x5 | RO | The status bit indicates that dbg power status |
| Note : this is a status bit. |
| 0x0000 : initial |
| 0x0001 : initial prun |
| 0x0010 : prun |
| 0x0011 : run |
| 0x0100 : poff |
| 0x0101 : off |
| [15:7] | - | 0x0 | - | reserved |
| [6] | reg\_en\_dbg\_timeout | 0x0 | RW | reg\_int\_dbg\_timeout interrupt enable, assert gpc\_int\_n if reg\_int\_dbg\_timeout is 1 |
| [5] | reg\_en\_dbg\_pup | 0x0 | RW | reg\_int\_dbg\_pup interrupt enable, assert gpc\_int\_n if reg\_int\_dbg\_pup is 1 |
| [4] | reg\_en\_dbg\_pdn | 0x0 | RW | reg\_int\_dbg\_pdn interrupt enable, assert gpc\_int\_n if reg\_int\_dbg\_pdn is 1 |
| [3] | - |  |  | reserved |
| [2] | reg\_int\_dbg\_timeout | 0x0 | RW1C | NoC idle sequence timeout status |
| [1] | reg\_int\_dbg\_pup | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power up sequence has done  0x1 : domain was powered up for the previous request |
| [0] | reg\_int\_dbg\_pdn | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power down sequence has done  0x1 : domain was powered down for the previous request |

CPU\_PCR: SYS domain power control register

Address: PMU Base + offset (0x60)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:2] | - | 0x0 | RO | reserved |
| [1] | reg\_cpu\_ccr | 0x0 | RW | Clock down enable  0x0 do not switch off clock even if ‘WFI’ is asserted  0x1 switch off clock when ‘WFI’ is assert |
| [0] | reg\_cpu\_pcr | 0x0 | RW | Power down enable:  0x0 do not switch off power even if ‘WFI’ is asserted.  0x1 Switch off power when ‘WFI’ is asserted.( must switch off clock ) |

CPU\_PUPSCR: SYS domain power up sequence control register

Address: PMU Base + offset (0x64)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_cpu\_sw2iso | 0xfff | RW | These bits define clk24m cycle counts from ‘power switch-ON’ to ‘isolation removal’.  **Note**: SW2ISO must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | reserved |
| [5:0] | reg\_cpu\_up\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-up request assertion (pup\_req)’ to ‘power up sequence’. |

CPU\_PDNSCR: SYS domain power down sequence control register

Address: PMU Base + offset (0x68)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | reg\_cpu\_iso2sw | 0xfff | RW | These bits define clk24m cycle counts from ‘isolation assertion’ to ‘power switch-OFF’.  **Note**: ISO2SW must not be programmed to zero. |
| [7:6] | - | 0x0 | RO | Reserved |
| [5:0] | reg\_cpu\_dn\_cnt | 0xf | RW | These bits define clk24m cycle counts from ‘power-down request assertion (pdn\_req)’ to ‘power down sequence’. |

CPU\_NOCSCR: SYS domain NIU sequence control register

Address: PMU Base + offset (0x6C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:8] | - | 0x0 | RO | reserved |
| [7:0] | reg\_cpu\_noc\_timeout | 0xff | RW | These bits define clk24m cycle counts from ‘noc idle request assertion’ to ‘noc idle timeout’, if noc idle timeout, gpc.pgc\_cpu0 enter poff directly |

CPU\_PSR: SYS domain power status register

Address: PMU Base + offset (0x70)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Name** | **Reset** | **Type** | **Description** |
| [31:24] | - | 0x0 | - | reserved |
| [23:20] | reg\_gpc\_cpu\_sts | 0x8 | RO | The status bit indicates that sys performance status |
| 0x0000 : PUP |
| 0x0001 : CUP |
| 0x0010 : WAITACTIVE |
| 0x0011 : RUN |
| 0x0100 : DNREQ |
| 0x0101 : WAITIDLE |
| 0x0110 : COFF |
| 0x0111 : PDN |
| 0x1000 : POFF |
| 0x1001 : OSCDN |
| 0x1010 : OSCOFF |
| 0x1011 : UPREQ |
| 0x1100 : OSCUP |
| 0x1101 : OSCACTIVE |
| [19:16] | reg\_pgc\_cpu\_sts | 0x5 | RO | The status bit indicates that sys power status |
| Note : this is a status bit. |
| 0x0000 : initial |
| 0x0001 : initial prun |
| 0x0010 : prun |
| 0x0011 : run |
| 0x0100 : poff |
| 0x0101 : off |
| [15:7] | - | 0x0 | - | reserved |
| [6] | reg\_en\_cpu\_timeout | 0x0 | RW | reg\_int\_cpu\_timeout interrupt enable, assert gpc\_int\_n if reg\_int\_cpu\_timeout is 1 |
| [5] | reg\_en\_cpu\_pup | 0x0 | RW | reg\_int\_cpu\_pup interrupt enable, assert gpc\_int\_n if reg\_int\_cpu\_pup is 1 |
| [4] | reg\_en\_cpu\_pdn | 0x0 | RW | reg\_int\_cpu\_pdn interrupt enable, assert gpc\_int\_n if reg\_int\_cpu\_pdn is 1 |
| [3] | - |  |  | reserved |
| [2] | reg\_int\_cpu\_timeout | 0x0 | RW1C | NoC idle sequence timeout status |
| [1] | reg\_int\_cpu\_pup | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power up sequence has done  0x1 : domain was powered up for the previous request |
| [0] | reg\_int\_cpu\_pdn | 0x0 | RW1C | Interrupt status: GPC hardware sets it as soon as the domain’s power down sequence has done  0x1 : domain was powered down for the previous request |

1. ISRAM Controller


5. 1. Register Summary

Table 151.ISRAM Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| memc\_status | 0x000 | 0x00000000 | Provides information about the configuration and current state of the SMC. |
| memif\_cfg | 0x004 | 0x00000000 | Provides information about the configuration of the memory interface |
| mem\_cfg\_set | 0x008 | - | Use this to:  • enable interrupts  • request the SMC to enter the Low-power state. |
| mem\_cfg\_clr | 0x00C | -- | Use this to:  • disable interrupts  • request the SMC to exit the Low-power state |
| direct\_cmd | 0x010 | -- | Initializes and updates the external memory devices using the data in the:  •Contains configuration data for the external memory devices. The data is  written to a memory device when the SMC receives a write to the Direct Command Register  •This write-only register is the holding register for the opmode[x]\_[n]  Registers. It contains configuration data for the external memory devices.  The data is written to a memory device when the SMC receives a write to  the Direct Command Register |
| set\_cycles | 0x014 | - | Contains configuration data for the external memory devices. The data is  written to a memory device when the SMC receives a write to the Direct  Command Register |
| set\_opmode | 0x018 | - | This write-only register is the holding register for the opmode[x]\_[n]  Registers. It contains configuration data for the external memory devices.  The data is written to a memory device when the SMC receives a write to  the Direct Command Register |
| refresh\_0 | 0x020 | 0x00000000 | Controls the insertion of idle cycles during consecutive bursts. This  enables PSRAM devices on memory interface 0 to initiate a refresh cycle |
| refresh\_1 | 0x024 | 0x00000000 | Controls the insertion of idle cycles during consecutive bursts. This  enables PSRAM devices on memory interface 1 to initiate a refresh cycle |
| sram\_cycles0\_0 | 0x100 | 0x0002B3CC | Returns the programmed timing parameters for SRAMs that connect to  memory interface [x] and chip select [n] |
| sram\_cycles0\_1 | 0x120 | 0x0002B3CC |
| sram\_cycles0\_2 | 0x140 | 0x0002B3CC |
| sram\_cycles0\_3 | 0x160 | 0x0002B3CC |
| sram\_cycles1\_0 | 0x180 | 0x0002B3CC |
| sram\_cycles1\_1 | 0x1A0 | 0x0002B3CC |
| sram\_cycles1\_2 | 0x1C0 | 0x0002B3CC |
| sram\_cycles1\_3 | 0x1E0 | 0x0002B3CC |
| opmode0\_0 | 0x104 | 0x00FF0802 | Returns the programmed operating mode for memory devices that connect  to memory interface [x] and chip select [n] |
| opmode0\_1 | 0x124 | 0x00FF0802 |
| opmode0\_2 | 0x144 | 0x00FF0802 |
| opmode0\_3 | 0x164 | 0x00FF0802 |
| opmode1\_0 | 0x184 | 0x00FF0802 |
| opmode1\_1 | 0x1A4 | 0x00FF0802 |
| opmode1\_2 | 0x1C4 | 0x00FF0802 |
| opmode1\_3 | 0x1E4 | 0x00FF0802 |
| user\_status | 0x200 | 0x0 | Provides the status of the user\_status7:0 inputs |
| user\_config | 0x204 | - | Controls the state of the user\_config7:0 outputs. |
| int\_cfg | 0xE00 | 0x0 | Controls the enabling of the integration test logic. |
| int\_inputs | 0xE04 | - | Provides the status of the following inputs:  • async1:0, msync1:0  • ebibackoff1:0, ebigrant1:0, and use\_ebi  • csysreq |
| int\_outputs | 0xE08 | - | Enables an external master to control the state of the following outputs:  • cactive  • csysack  • ebireq1:0  • smc\_int, smc\_int1:0 |
| periph\_id\_n | 0xFE0-0xFEC | 0x00\_4135\_c | Provide information about the configuration and version of the peripheral. |
| pcell\_id\_n | 0xFF0-0xFFC | 0xB105F00D | When concatenated, these four registers return 0xB105F00D to indicate that  the SMC is a CoreLink peripheral. |

Notes:

Bits1:0 and 31:16 are dependent on external tie-offs. The remaining bits default to 0.

* 1. Register Descriptions

nemc\_status: memc\_status Register Descriptions

Address: Operational Base + offset (0x000)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:13 | RO | 0x0 | Reserved, read undefined |
| 12 | RO | 0x0 | Raw status of the ecc\_int1 interrupt signal. |
| 11 | RO | 0x0 | Raw status of the ecc\_int0 interrupt signal. |
| 10 | RO | 0x0 | Status of the ecc\_int1 interrupt signal after ANDing with its enable bit, ecc\_int\_en1. |
| 9 | RO | 0x0 | Status of the ecc\_int0 interrupt signal after ANDing with its enable bit, ecc\_int\_en0. |
| 8 | RO | 0x0 | Interrupt enable status for ecc\_int1:  0 = Interrupt is disabled so ecc\_int1 is LOW.  1 = Interrupt is enabled. |
| 7 | RO | 0x0 | Interrupt enable status for ecc\_int0:  0 = Interrupt is disabled so ecc\_int0 is LOW.  1 = Interrupt is enabled. |
| 6 | RO | 0x0 | Raw status of the smc\_int1 interrupt signal. |
| 5 | RO | 0x0 | Raw status of the smc\_int0 interrupt signal. |
| 4 | RO | 0x0 | Status of the smc\_int1 interrupt signal after ANDing with its enable bit, int\_en1. |
| 3 | RO | 0x0 | Status of the smc\_int0 interrupt signal after ANDing with its enable bit, int\_en0. |
| 2 | RO | 0x0 | Interrupt enable status for smc\_int1:  0 = Interrupt is disabled so smc\_int1 is LOW.  1 = Interrupt is enabled. |
| 1 | RO | 0x0 | Interrupt enable status for smc\_int0:  0 = Interrupt is disabled so smc\_int0 is LOW.  1 = Interrupt is enabled. |
| 0 | RO | 0x0 | Operating state of the SMC:  0 = SMC is in the Ready state  1 = SMC is in the Low-power state. |

Note:

Provides information about the configuration and current state of the Isram controller

semif\_cfg: memif\_cfg Register Descriptions

Address: Operational Base + offset (0x004)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:18 | RO | 0x0 | Reserved, write as zero. |
| 17:16 | RO | 0x3 | Returns the number of exclusive access monitor resources that are implemented in the SMC.  B00 = 0 monitors  B01 = 1 monitor  B10 = 2 monitors  B11 = 3 monitors |
| 15 | RO | 0x0 | Reserved |
| 14 | RO | 0x0 | Returns the value of the remap\_1 input. |
| 13:12 | RO | 0x2 | Returns the maximum width of the SMC memory data bus for interface 1:  B00 = 8 bits  B01 = 16 bits  B10 = 32 bits  B11 = reserved |
| 11:10 | RO | 0x0 | Returns the number of different chip selects that the memory interface 1 supports:  B00 = 1 chip  B01 = 2 chips  B10 = 3 chips  B11 = 4 chips |
| 9:8 | RO | 0x1 | Returns the memory interface 1 type:  B00 = configuration does not include this memory interface  B01 = SRAM non-multiplexed  B10 = NAND  B11 = SRAM multiplexed  If b00, the remaining bit slices for memory interface 1 are always read as 0 |
| 7 | RO | 0x0 | Reserved |
| 6 | RO | 0x0 | Returns the value of the remap\_0 input. |
| 5:4 | RO | 0x2 | Returns the maximum width of the SMC memory data bus for interface 0:  B00 = 8 bits  B01 = 16 bits  B10 = 32 bits  B11 = reserved |
| 3:2 | RO | 0x0 | Returns the number of different chip selects that the memory interface 0 supports:  B00 = 1 chip  B01 = 2 chips  B10 = 3 chips  B11 = 4 chips |
| 1:0 | RO | 0x1 | Returns the memory interface 0 type:  B00 = reserved  B01 = SRAM non-multiplexed  B10 = NAND  B11 = SRAM multiplexed |

Note:

Provide information about the configuration of the memory interface.

mem\_cfg\_set: mem\_cfg\_set Register Descriptions

Address: Operational Base + offset (0x008)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:7 | WO | - | Reserved, write as zero. |
| 6 | WO | - | 0 = No effect  1 = Enables the ecc\_int1 interrupt. SMC sets the ecc\_int\_en1 bit to 1 in the memc\_status Register |
| 5 | WO | - | 0 = No effect  1 = Enables the ecc\_int0 interrupt. SMC sets the ecc\_int\_en0 bit to 1 in the memc\_status Register |
| 4:3 | WO | - | Reserved, write as zero. |
| 2 | WO | - | 0 = No effect  1 = Requests the SMC to enter Low-power state when it next becomes idle. |
| 1 | WO | - | 0 = No effect  1 = Enables the smc\_int1 interrupt. SMC sets the int\_en1 bit to 1 in the memc\_status Register |
| 0 | WO | - | 0 = No effect  1 = Enables the smc\_int0 interrupt. SMC sets the int\_en0 bit to 1 in the memc\_status Register, |

Note:

Enable interrupts, request the SMC to enter the Low-power state.

mem\_cfg\_clr : mem\_cfg\_clr Register Descriptions

Address: Operational Base + offset (0x00c)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:7 | WO | - | Reserved, write as zero. |
| 6 | WO | - | 0 = No effect  1 = Enables the ecc\_int1 interrupt. SMC sets the ecc\_int\_en1 bit to 1 in the memc\_status Register |
| 5 | WO | - | 0 = No effect  1 = Enables the ecc\_int0 interrupt. SMC sets the ecc\_int\_en0 bit to 1 in the memc\_status Register |
| 4 | WO | - | 0 = No effect  1 = Clear SMC Interrupt 1, as an alternative to an AXI read. |
| 3 | WO | - | 0 = No effect  1 = Clear SMC Interrupt 0, as an alternative to an AXI read. |
| 2 | WO | - | 0 = No effect  1 = Requests the SMC to enter Low-power state when it next becomes idle. |
| 1 | WO | - | 0 = No effect  1 = Enables the smc\_int1 interrupt. SMC sets the int\_en1 bit to 1 in the memc\_status Register |
| 0 | WO | - | 0 = No effect  1 = Enables the smc\_int0 interrupt. SMC sets the int\_en0 bit to 1 in the memc\_status Register, |

Note:

Disable interrupts.

direct\_cmd: direct\_cmd Register Descriptions

Address: Operational Base + offset (0x010)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:26 | WO | - | Reserved, write as zero. |
| 25:23 | WO | - | Selects chip configuration register bank to update, and enables chip mode register access  depending on cmd\_type. The encoding is:  b000-b011 = Chip selects 1-4 on memory interface 0  b100-b111 = Chip selects 1-4 on memory interface 1. |
| 22:21 | WO | - | Selects the command type:  b00 = UpdateRegs and AXI  b01 = ModeReg  b10 = UpdateRegs  b11 = ModeReg and UpdateRegs. |
| 20 | WO | - | Maps to the configuration register enable signal, cre, when a ModeReg command is issued.  The encoding is:  0 = cre is LOW  1 = cre is HIGH when ModeReg write occurs. |
| 19:0 | WO | - | When cmd\_type = UpdateRegs and AXI then:  • bits 15:0 are used to match wdata15:0  • bits 19:16 are reserved. Write as zero.  When cmd\_type = ModeReg or ModeReg and UpdateRegs, these bits map to the external  memory address bits 19:0.  When cmd\_type = UpdateRegs, these bits are reserved. Write as zero. |

Note:

Initializes and updates the external memory devices.

set\_cycles: set\_cycles Register Descriptions

Address: Operational Base + offset (0x014)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | WO | - | Reserved, write as zero. |
| 23:20 | WO | - | Contains the value to be written to either the:  • we\_time bit of the SRAM Cycles Register on |
| 19:17 | WO | - | Contains the value to be written to either the:  • t\_tr field of the SRAM Cycles Register |
| 16:14 | WO | - | set\_t4 Contains the value to be written to either the:  • t\_pc field of the SRAM Cycles Register |
| 13:11 | WO | - | Contains the value to be written to the t\_wp field in either the:  • SRAM Cycles Register |
| 10:8 | WO | - | Contains the value to be written to either the:  • t\_ceoe field of the SRAM Cycles Register |
| 7:4 | WO | - | Contains the value to be written to the t\_wc field in either the:  • SRAM Cycles Register |
| 3:0 | WO | - | set\_t0 Contains the value to be written to the t\_rc field in either the:  • SRAM Cycles Register |
| 31:16 | WO | - | Reserved, write as zero. |
| 15:13 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register burst\_align  field.  When you configure the SMC to perform synchronous transfersa, these bits control if  memory bursts are split on memory burst boundaries:  b000 = bursts can cross any address boundary  b001 = burst split on memory burst boundary, that is, 32 beats for continuous  b010 = burst split on 64 beat boundary  b011 = burst split on 128 beat boundary  b100 = burst split on 256 beat boundary  b101-b111 = reserved. |
| 12 | WO | - | set\_bls\_time Contains the value to be written to the specific SRAM chip opmode Register byte lane strobe  (bls) bit.This bit affects the assertion of the byte lane strobe outputs.  0 = bls timing equals chip select timing. This is the default setting.  1 = bls timing equals we\_n timing. This setting is used for eight memories that have no bls\_n  Inputs. In this case, the bls\_n output of the SMC is connected to the we\_n memory input. |
| 11 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register address valid  (adv) bit. The memory uses the address advance signal adv\_n when set. |
| 10 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register burst address  advance (baa) bit. The memory uses the baa\_n signal when set. |
| 9:7 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register wr\_bl field.  Encodes the memory burst length:  b000 = 1 beat  b001 = 4 beats  b010 = 8 beats  b011 = 16 beats  b100 = 32 beats  b101 = continuous  b110-b111 = reserved. |
| 6 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register wr\_sync bit.  The memory writes are synchronous when set. |
| 5:3 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register rd\_bl field.  Encodes the memory burst length:  b000 = 1 beat  b001 = 4 beats  b010 = 8 beats  b011 = 16 beats  b100 = 32 beats  b101 = continuous  b110-b111 = reserved. |
| 2 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register rd\_sync bit.  Memory in sync mode when set. |
| 1:0 | WO | - | Contains the value to be written to the specific chip opmode Register memory width (mw)  field.  Encodes the memory data bus width:  b00 = 8 bitsb  b01 = 16 bitsb  b10 = 32 bits  b11 = reserved.  You can program this to the configured width, or half that width. See Memory Interface Configuration Register |

Note:

Contains configuration data for the external memory devices.

set\_opmode: set\_opmode Register Descriptions

Address: Operational Base + offset (0x018)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | WO | - | Reserved, write as zero. |
| 15:13 | WO | - | Contains the value to be written to the specific SRAMchip opmode Register burst\_align  field.  When you configure the SMC to perform synchronous transfersa  , these bits control if  memory bursts are split on memory burst boundaries:  b000 = bursts can cross any address boundary  b001 = burst split on memory burst boundary, that is, 32 beats for continuous  b010 = burst split on 64 beat boundary  b011 = burst split on 128 beat boundary  b100 = burst split on 256 beat boundary  b101- b111 = reserved. |
| 12 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register byte lane strobe  (bls) bit. This bit affects the assertion of the byte lane strobe outputs.  0 = bls timing equals chip select timing. This is the default setting.  1 = bls timing equals we\_ntiming. This setting is used for eight memories that have no bls\_n  inputs. In this case, the bls\_noutput of the SMC is connected to the we\_nmemory input. |
| 11 | WO | - | Contains the value tobe written to the specificSRAM chip opmode Register address valid  (adv) bit. The memory usesthe address advance signal adv\_nwhen set. |
| 10 | WO | - | Contains the value tobe written to the specificSRAM chip opmode Register burst address  advance(baa) bit. The memory uses the baa\_nsignal when set. |
| 9:7 | WO | - | Contains the value tobe written to the specific SRAMchip opmode Register wr\_bl field.  Encodes the memory burst length:  b000 = 1 beat  b001 = 4 beats  b010 = 8 beats  b011 = 16 beats  b100 = 32 beats  b101 = continuous  b110-b111 = reserved |
| 6 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register wr\_sync bit.  The memory writes are synchronous when set. |
| 5:3 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register rd\_bl field.  Encodes the memory burst length:  b000 = 1 beat  b001 = 4 beats  b010 = 8 beats  b011 = 16 beats  b100 = 32 beats  b101 = continuous  b110-b111 = reserved. |
| 2 | WO | - | Contains the value to be written to the specific SRAM chip opmode Register rd\_sync bit.  Memory in sync mode when set. |
| 1:0 | WO | - | Contains the value to be written to the specific chip opmode Register memory width(mw)  field.  Encodes the memory data bus width:  b00 = 8 bits  b  b01 = 16 bits  b  b10 = 32 bits  b11 = reserved. |

Note:

This write-only register is the holding register for the opmode[x]\_[n] Registers.

refresh\_0: refresh\_0 Register Descriptions

Address: Operational Base + offset (0x020)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | RW | 0 | Reserved, write as zero. |
| 3:0 | RW | 0 | Sets the number of consecutive memory burstsa that the SMC permits, on memory interface 0,  before it deasserts the chip select. The options are:  b0000 = disables the insertion of idle cycles between consecutive bursts  b0001 = an idle cycle occurs after each burst  b0010 = an idle cycle occurs after 2 consecutive bursts  b0011 = an idle cycle occurs after 3 consecutive bursts  b0100 = an idle cycle occurs after 4 consecutive bursts  b1111 = an idle cycle occurs after 15 consecutive bursts. |

Note:

Controls the insertion of idle cycles during consecutive bursts. This enables PSRAM devices on memory interface 0 to initiate a refresh cycle.

refresh\_1: refresh\_1 Register Descriptions

Address: Operational Base + offset (0x024)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | RW | 0 | Reserved, write as zero. |
| 3:0 | RW | 0 | Sets the number of consecutive memory burstsa that the SMC permits, on memory interface 0,  before it deasserts the chip select. The options are:  b0000 = disables the insertion of idle cycles between consecutive bursts  b0001 = an idle cycle occurs after each burst  b0010 = an idle cycle occurs after 2 consecutive bursts  b0011 = an idle cycle occurs after 3 consecutive bursts  b0100 = an idle cycle occurs after 4 consecutive bursts  ……  b1111 = an idle cycle occurs after 15 consecutive bursts. |

Note:

Controls the insertion of idle cycles during consecutive bursts. This enables PSRAM devices on memory interface 1 to initiate a refresh cycle.

The sram\_cycles[x]\_[n]: The sram\_cycles[x]\_[n] Register Descriptions

Address: Operational Base + offset (0x100、0x120、0x140、0x160、0x180、0x1A0、0x1C0、0x1E0)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:21 | RO | 0x0 | Reserved, read undefined. |
| 20 | RO | 0x0 | For asynchronous multiplexed transfers this bit returns when the SMC asserts we\_n:  0 = SMC asserts we\_n two mclk cycles after asserting cs\_n  1 = SMC asserts we\_n and cs\_n together. |
| 19:17 | RO | 0x1 | Returns the turnaround time. Minimum permitted value = 1. |
| 16:14 | RO | 0x2 | Returns the page cycle time. Minimum permitted value = 1. |
| 13:11 | RO | 0x6 | Returns the we\_n assertion delay. Minimum permitted value = 1. |
| 10:8 | RO | 0x3 | Returns the oe\_n assertion delay. Minimum permitted value = 1. |
| 7:4 | RO | 0xC | Returns the write cycle time. Minimum permitted value = 2. |
| 3:0 | RO | 0xC | Returns the read cycle time. Minimum permitted value = 2. |

Note:

Returns the programmed timing parameters for SRAMs that connect to memory interface [x] and chip select [n].

opmode[x]\_[n]: opmode[x]\_[n] Register Descriptions

Address: Operational Base + offset (0x104、0x124、0x144、0x164、0x184、0x1A4、0x1C4、0x1E4)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | WO | - | Function |
| 23:16 | WO | -t | Returns the value of the addr\_match[x]\_[n]7:0 tie-off. This is the comparison value  for address bits 31:24 to determine the chip that is selected |
| 15:13 | WO | - | Returns the value of the addr\_mask[x]\_[n]7:0 tie-off. This is the mask for address  bits31:24. A logic 1 indicates the bit is used for comparison. |
| 12 | WO | - | When you program the SMC to perform synchronous transfersa, these bits return the  memory burst operating mode:  b000 = bursts can cross any address boundary. This is the default setting.  b001 = burst split on memory burst boundary, that is, 32 beats for continuous.  b010 = burst split on 64 beat boundary.  b011 = burst split on 128 beat boundary.  b100 = burst split on 256 beat boundary.  b101-b111 = reserved.  The reset value is b000. |
| 11 | WO | - | Returns the byte lane strobe operating mode for an SRAM memory interface:  0 = bls timing equals chip select timing. This is the default setting.  1 = bls timing equals we\_n timing. |
| 10 | WO | - | Returns the address advance signal operating mode for an SRAM memory interface:  0 = SMC ties adv\_n HIGH. This is the default setting.  1 = SMC sets adv\_n LOW at the start of a transfer. |
| 9:7 | WO | - | Returns the burst address advance signal operating mode for an SRAM memory  interface:  0 = SMC ties baa\_n HIGH. This is the default setting.  1 = SMC sets baa\_n LOW |
| 6 | WO | - | Returns the memory burst length for writes on an SRAM memory interface:  b000 = 1 beat. This is the default setting.  b001 = 4 beats  b010 = 8 beats  b011 = 16 beats  b100 = 32 beats  b101 = continuous  b110-b111 = reserved. |
| 5:3 | WO | - | Returns the write operating mode for an SRAM memory interface:  0 = SMC performs asynchronous writes. This is the default setting.  1 = SMC performs synchronous writes. |
| 2 | WO | - | Returns the memory burst length for reads on an SRAM memory interface:  b000 = 1 beat. This is the default setting.  b001 = 4 beats  b010 = 8 beats  b011 = 16 beats  b100 = 32 beats  b101 = continuous  b110-b111 = reserved. |
| 1:0 | WO | - | Returns the read operating mode for an SRAM memory interface:  0 = SMC performs asynchronous reads. This is the default setting.  1 = SMC performs synchronous reads. |

Note:

Returns the programmed operating mode for memory devices that connect to memory interface [x] and chip select [n].

user\_status: user\_status Register Descriptions

Address: Operational Base + offset (0x200)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined |
| 7:0 | RO | 0x0 | This value returns the state of the user\_status7:0 inputs |

Note:Provides the status of the user\_status7:0inputs.

user\_config: user\_config Register Descriptions

Address: Operational Base + offset (0x204)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | WO | - | Reserved, read undefined |
| 7:0 | WO | - | This value sets the state of the user\_config7:0 outputs |

Note:Controls the state of the user\_config7:0outputs

periph\_id: periph\_id Register Descriptions

Address: Operational Base + offset (0xFE0-0xFEC)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:25 | RO | 0x0 | Reserved, read undefined |
| 24 | RO | 0x0 | Configuration option is peripheral-specific. See Peripheral Identification Register 3 |
| 23:20 | RO | 0x6 | Identifies the RTL revision of the peripheral. |
| 19:12 | RO | 0x41 | Identifies the designer. This is 0x41 for ARM. |
| 11:0 | RO | 0x354 | Identifies the peripheral. The part numbers for the SMC are:  • 0x354 for SMC-354. |

Note:Provide information about the configuration and version of the peripheral.

periph\_id\_0: periph\_id\_0 Register Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined |
| 7:0 | RO | 0x54 | Returns 0x5x |

Note:

Register is hard-coded and the fields in the register determine the reset value

periph\_id\_1: periph\_id\_1 Register Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined |
| 7:4 | RO | 0x1 | Returns 0x1 |
| 3:0 | RO | 0x3 | Returns 0x3 |

Note:Register is hard-coded and the fields in the register determine the reset value

The periph\_id\_2: periph\_id\_2 Register Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined |
| 7:4 | RO | 0x6 | These bits read back as:  • 0x1 for r1p0  • 0x2 for r1p1  • 0x3 for r1p2  • 0x4 for r2p0  • 0x5 for r2p1  • 0x6 for r2p2 |
| 3:0 | RO | 0x4 | Returns 0x4 |

Note:Register is hard-coded and the fields in the register determine the reset value

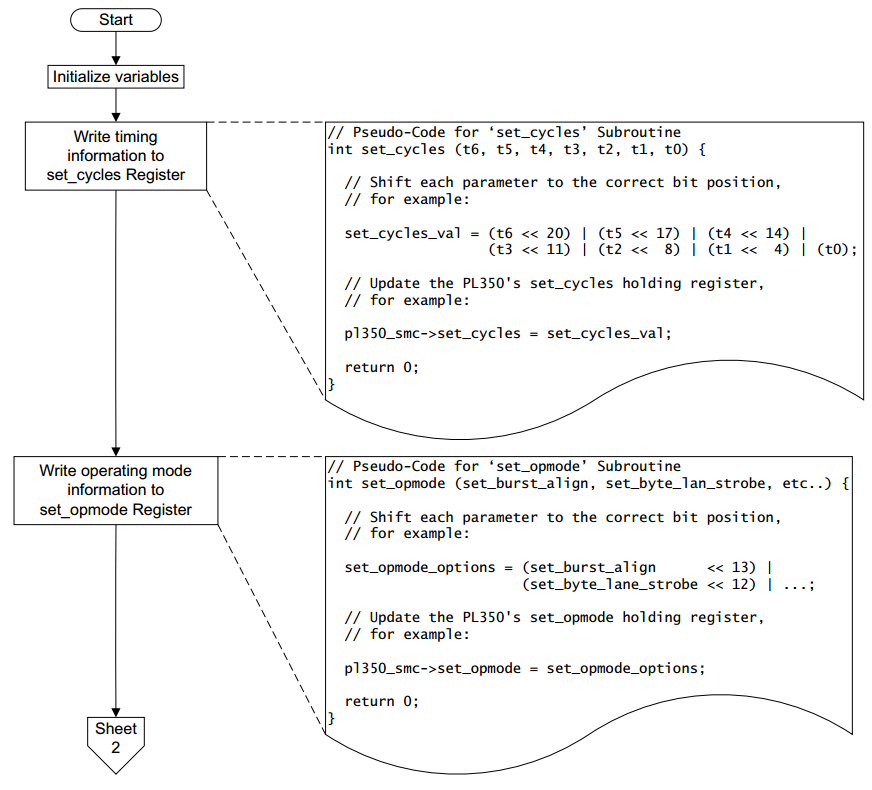
pcell\_id\_3:0 Register Descriptions

Address: Operational Base + offset (0xFF0-0xFFC)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RO | 0xB1 | 31:8 Reserved, read undefined |
| 7:0 Returns 0xB1 |
| 23:16 | RO | 0x05 | 31:8 Read undefined |
| 7:0 Returns 0x05 |
| 15:8 | RO | 0xF0 | 31:8 Read undefined |
| 7:0 Returns 0xF0 |
| 7:0 | RO | 0x0D | 31:8 Read undefined |
| 7:0 Returns 0x0D |

Note：When concatenated, these four registers return 0xB105F00Dto indicate that the SMC is a CoreLink peripheral.

* 1. Programming Sequence



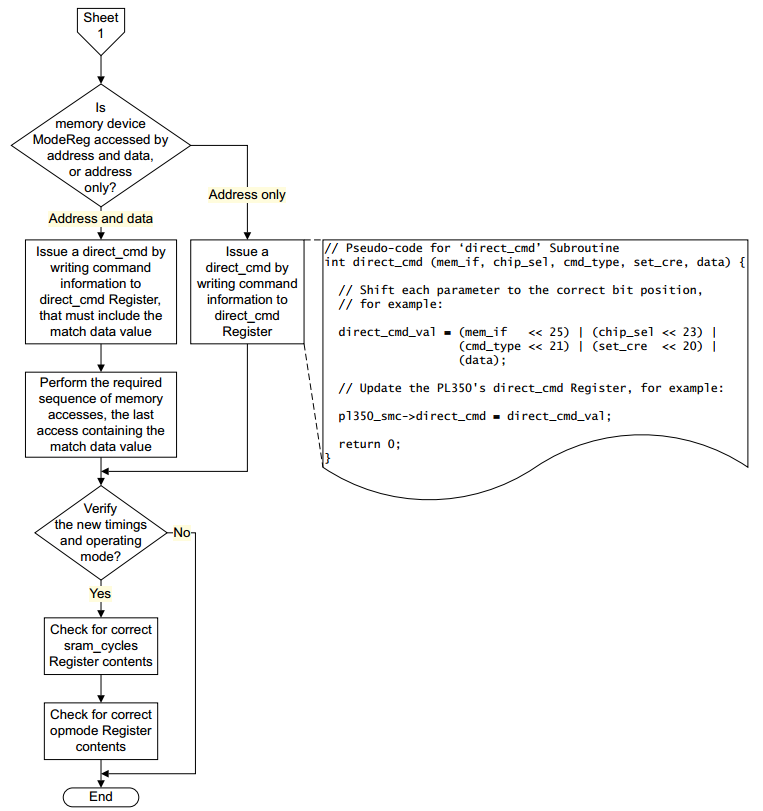


Figure 2.Transmit programming

Sequence of events that a device driver must carry out to initialize the SMC and a memory device to ensure the configuration of both is synchronized.

Typically, PSRAM devices can have the mode register programmed using the address bus only. NOR flash memory devices are examples of memory that require mode register accesses to be carried out using a sequence of accesses using the address and data buses. Check the data sheet for the specific memory device you are configuring to determine the configuration method.

1. CoreSight
2. 1. Register Summary
      1. APB Interconnect Register Summary

Table 152.APB Interconnect Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| CROM\_ENTRY | 0x0FC | UNKNOWN | ROM Table |
| PIDR4 | 0xFD0 | UNKNOWN | Peripheral ID4Register |
| PIDR0 | 0xFE0 | UNKNOWN | Peripheral ID0 Register |
| PIDR1 | 0xFE4 | UNKNOWN | Peripheral ID1 Register |
| PIDR2 | 0xFE8 | UNKNOWN | Peripheral ID2 Register |
| PIDR3 | 0xFEC | 0x00000000 | Peripheral ID3 Register |
| CIDR0 | 0xFF0 | 0x0000000D | Component ID0Register |
| CIDR1 | 0xFF4 | 0x00000010 | Component ID1 Register |
| CIDR2 | 0xFF8 | 0x00000005 | Component ID2 Register |
| CIDR3 | 0xFFC | 0x000000B1 | Component ID3 Register |

* + 1. APB-AP Register Summary

Table 153.APB-AP Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| CSW, | 0x00 | 0x00000002 | ROM Table |
| TAR | 0x04 | 0x00000000 | Peripheral ID4Register |
| DRW | 0x0c | 0x00000000 | Peripheral ID0 Register |
| BD0 | 0x10 | UNKNOWN | Reserved |
| BD1 | 0x14 | UNKNOWN | Reserved |
| BD2 | 0x18 | UNKNOWN | Reserved |
| BD3 | 0x1C | UNKNOWN | Reserved |
| BASE | 0xF8 | 0x80000003 | Peripheral ID1 Register |
| IDR | 0xFC | 0x44770002 | Peripheral ID2 Register |

* + 1. Debug port register summary

Table 7.Debug port Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| ABORT, | N/A | 0x00000000 | AP Abort Register |
| IDCODE | N/A | 0x6BA00877 | ID Code Register |
| DPIDR | N/A | 0x6BA02877 | Debug Port Identification Register |
| CTRL/STAT | N/A | 0x00000000 | Control/Status Register |
| SELECT | N/A | 0x00000000 | AP Select Register |
| RDBUFF | N/A | 0x00000000 | Read Buffer Register. |
| DLCR | N/A | 0x00000060 | Data Link Control Register |
| TARGETID | N/A | 0x088C001D | Target Identification Register |
| DLPIDR | N/A | 0x00000000 | Data Link Protocol Identification Register |
| RESEND | N/A | 0x00000000 | Read Resend Register. |

* + 1. CTI Register Summary

Table 154. CTI Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| CTICONTROL | 0x000 | 0x00000000 | The CTI Control Register enables the CTI. |
| CTIINTACK | 0x010 | 0x00000000 | The CTI Interrupt Acknowledge Register is WO. Any bits written as a 1 cause the CTITRIGOUT output signal to be acknowledged. The acknowledgement is cleared when MAPTRIGOUT is deactivated. This register is used when the CTITRIGOUT is used as a sticky output, that is, no hardware acknowledge is supplied, and a software acknowledge is required. |
| CTIAPPSET | 0x014 | 0x00000000 | The CTI Application Trigger Set Register is RW. A write to this register causes a channel event to be raised, corresponding to the bit written to. |
| CTIAPPCLEAR | 0x018 | 0x00000000 | The CTI Interrupt Acknowledge Register is WO. A write to this register causes a channel event to be cleared, corresponding to the bit written to. |
| CTIAPPPULSE | 0x01C | 0x00000000 | The CTI Application Pulse Register is write-only. A write to this register causes a channel event pulse, one CTICLK period, to be generated, corresponding to the bit written to. The pulse external to the ECT can be extended to multi-cycle by the handshaking interface circuits. This register clears itself immediately, so it can be repeatedly written to without software having to clear it. |
| CTIINEN0 | 0x020 | 0x00000000 | The CTI Trigger 0 to Channel Enable Register enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN1 | 0x024 | 0x00000000 | The CTI Trigger 1 to Channel Enable Register enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN2 | 0x028 | 0x00000000 | The CTI Trigger to Channel Enable Register 0 enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN3 | 0x02C | 0x00000000 | The CTI Trigger to Channel Enable Register 0 enables the signaling of an ever issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN4 | 0x030 | 0x00000000 | The CTI Trigger to Channel Enable Register 0 enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN5 | 0x034 | 0x00000000 | The CTI Trigger to Channel Enable Register 0 enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN6 | 0x038 | 0x00000000 | The CTI Trigger to Channel Enable Register 0 enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIINEN7 | 0x03C | 0x00000000 | The CTI Trigger to Channel Enable Register 0 enables the signaling of an event on CTM channels when the core issues a trigger, CTITRIGIN, to the CTI. Within this register there is one bit for each of the four channels implemented. This register does not affect the application trigger operations. |
| CTIOUTEN0 | 0x0A0 | 0x00000000 | The CTI Channel to Trigger 0 Enable Register define which channels can generate a CTITRIGOUT [0] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN1 | 0x0A4 | 0x00000000 | The CTI Channel to Trigger 1 Enable Register define which channels can generate a CTITRIGOUT [1] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN2 | 0x0A8 | 0x00000000 | The CTI Channel to Trigger 2 Enable Register define which channels can generate a CTITRIGOUT [2] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN3 | 0x0AC | 0x00000000 | The CTI Channel to Trigger 3 Enable Registers define which channels can generate a CTITRIGOUT [3] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN4 | 0x0B0 | 0x00000000 | The CTI Channel to Trigger 4 Enable Registers define which channels can generate a CTITRIGOUT [4] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN5 | 0x0B4 | 0x00000000 | The CTI Channel to Trigger 5 Enable Registers define which channels can generate a CTITRIGOUT [5] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN6 | 0x0B8 | 0x00000000 | The CTI Channel to Trigger 6 Enable Registers define which channels can generate a CTITRIGOUT [6] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTIOUTEN7 | 0x0BC | 0x00000000 | The CTI Channel to Trigger 7 Enable Registers define which channels can generate a CTITRIGOUT [7] output. Within this register there is one bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs. |
| CTITRIGINSTATUS | 0x130 | 0x00000000 | The CTI Trigger In Status Register provides the status of the CTITRIGIN inputs. |
| CTITRIGOUTSTATUS | 0x134 | 0x00000000 | The CTI Trigger Out Status Register provides the status of the CTITRIGOUT outputs. |
| CTICHINSTATUS | 0x138 | 0x00000000 | The CTI Channel In Status Register provides the status of the CTICHIN inputs. |
| CTICHOUTSTATUS | 0x13C | 0x00000000 | The CTI Channel Out Status Register provides the status of the CTI CTICHOUT outputs. |
| CTIGATE | 0x140 | 0x0000000F | The Gate Enable Register prevents the channels from propagating through the CTM to other CTIs. This enables local cross-triggering, for example for causing an interrupt when the ETM trigger occurs. It can be used effectively with CTIAPPSET, CTIAPPCLEAR, and CTIAPPPULSE for asserting trigger outputs by asserting channels, without affecting the rest of the system. On reset, this register is 0xF, and channel propagation is enabled. |
| ASICCTL | 0x144 | 0x00000000 | IMPLEMENTATION DEFINED ASIC control, value written to the register is output on ASICCTL [7:0]. |
| ITCHINACK | 0xEDC | 0x00000000 | This register is a WO register. It can be used to set the value of the CTCHINACK outputs. |
| ITTRIGINACK | 0xEE0 | 0x00000000 | This register is a WO register. It can be used to set the value of the CTTRIGINACK outputs |
| ITCHOUT | 0xEE4 | 0x00000000 | This register is a WO register. It can be used to set the value of the CTCHOUT outputs. |
| ITTRIGOUT | 0xEE8 | 0x00000000 | This register is a WO register. It can be used to set the value of the CTTRIGOUT outputs. |
| ITCHOUTACK | 0xEEC | 0x00000000 | This register is a read-only register. It can be used to read the values of the CTCHOUTACK inputs. |
| ITTRIGOUTACK | 0xEF0 | 0x00000000 | This register is a RO register. It can be used to read the values of the CTTRIGOUTACK inputs. |
| ITCHIN | 0xEF4 | 0x00000000 | This register is a RO register. It can be used to read the values of the CTCHIN inputs. |
| ITTRIGIN | 0xEF8 | 0x00000000 | This register is a RO register. It can be used to read the values of the CTTRIGIN inputs. |
| ITCTRL | 0xF00 | 0x00000000 | This register enables topology detection. This register enables the component to switch from a functional mode, the default behavior, to integration mode where the inputs and outputs of the component can be directly controlled for the purposes of integration testing and topology detection.  After performing integration or topology detection, you must reset the system to ensure correct behavior of CoreSight and other connected system components that the integration or topology detection can affect. |
| CLAIMSET | 0xFA0 | 0x0000000F | Software can use the claim tag to coordinate application and debugger access to trace unit functionality. The claim tags have no effect on the operation of the component. The CLAIMSET register sets bits in the claim tag, and determines the number of claim bits implemented. |
| CLAIMCLR | 0xFA4 | 0x00000000 | Software can use the claim tag to coordinate application and debugger access to trace unit functionality. The claim tags have no effect on the operation of the component. The CLAIMCLR Register clears the bits in the claim tag, and determines the current value of the claim tag. |
| LAR | 0xFB0 | 0x00000000 | Controls write access from self-hosted, on-chip accesses. The LAR Register does not affect the accesses using the external debugger interface. |
| LSR | 0xFB4 | 0x00000003 | Indicates the status of the lock control mechanism. This lock prevents accidental writes by code being debugged. When locked, write accesses are denied for all registers except for the LAR Register. The lock registers do not affect accesses from the external debug interface. This register reads as 0 when accessed from the external debug interface. |
| AUTHSTATUS | 0xFB8 | 0x00000005 | Reports the required security level and present status. |
| DEVID | 0xFC8 | 0x00040800 | Indicates the capabilities of the component |
| DEVTYPE | 0xFCC | 0x00000014 | Provides a debugger with information about the component when the Part Number field is not recognized. The debugger can then report this information. |
| PIDR4 | 0xFD0 | 0x00000004 | Part of the set of peripheral identification registers. Contains part of the designer identity and the memory size. |
| PIDR5 | 0xFD4 | 0x00000000 | Reserved. |
| PIDR6 | 0xFD8 | 0x00000000 | Reserved. |
| PIDR7 | 0xFDC | 0x00000000 | Reserved. |
| PIDR0 | 0xFE0 | 0x00000006 | Part of the set of peripheral identification registers. Contains part of the designer-specific part number. |
| PIDR1 | 0xFE4 | 0x000000B9 | Part of the set of peripheral identification registers. Contains part of the designer-specific part number and part of the designer identity. |
| PIDR2 | 0xFE8 | 0x0000003B | Part of the set of peripheral identification registers. Contains part of the designer identity and the product revision. |
| PIDR3 | 0xFEC | 0x00000000 | Part of the set of peripheral identification registers. Contains the REVAND and CMOD fields. |
| CIDR0 | 0xFF0 | 0x0000000D | A component identification register that indicates the identification registers are present. |
| CIDR1 | 0xFF4 | 0x00000090 | A component identification register that indicates the identification registers are present. This register also indicates the component class. |
| CIDR2 | 0xFF8 | 0x00000005 | A component identification register that indicates the identification registers are present. |
| CIDR3 | 0xFFC | 0x000000B1 | A component identification register that indicates the identification registers are present. |

* + 1. TPIU Register Summary

Table 155. TPIU Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| Supported\_Port\_Sizes | 0x000 | 0x00000001 | Each bit location represents a single port size that is supported on the device, that is, 32-1 in bit locations [31:0]. If the bit is set then that port size is permitted. By default the RTL is designed to support all port sizes, set to 0xFFFFFFFF. This register reflects the value of the CSTPIU\_SUPPORTSIZE\_VAL Verilog define value, currently not user-modifiable, and is more constrained by the input tie-off TPMAXDATASIZE. The external tie-off, TPMAXDATASIZE, must be set during finalization of the ASIC to reflect the actual number of TRACEDATA signals being wired to physical pins. This is to ensure that tools do not attempt to select a port width that cannot be captured by an attached Trace Port Analyzer (TPA). The value on TPMAXDATASIZE causes bits within the Supported Port Size register that represent wider widths to be clear, that is, unsupported. |
| Current\_port\_size | 0x004 | 0x00000001 | The Current Port Size Register has the same format as the Supported Port Sizes register but only one bit is set, and all others must be zero. Writing values with more than one bit set or setting a bit that is not indicated as supported is not supported and causes unpredictable behavior. On reset this defaults to the smallest possible port size, 1 bit, and so reads as 0x00000001. |
| Supported\_trigger\_modes | 0x100 | 0x0000011F | This register indicates the implemented trigger counter multipliers and other supported features of the trigger system. |
| Trigger\_counter\_value | 0x104 | 0x00000000 | The Trigger Counter Register enables delaying the indication of triggers to any external connected trace capture or storage devices. This counter is only eight bits wide and is intended to only be used with the counter multipliers in the Trigger Multiplier Register, 0x108. When a trigger is started, this value, in combination with the multiplier, is the number of words before the trigger is indicated. When the trigger counter reaches zero, the value written here is reloaded. Writing to this register causes the trigger counter value to reset but not reset any values on the multiplier. Reading this register returns the preset value not the current count. |
| Trigger\_multiplier | 0x108 | 0x00000000 | This register contains the selectors for the trigger counter multiplier. Several multipliers can be selected to create the required multiplier value, that is, any value between one and approximately 2x10^9. The default value is multiplied by one, 0x0. Writing to this register causes the internal trigger counter and the state in the multipliers to be reset to initial count position, that is, trigger counter is reloaded with the Trigger Counter Register value and all multipliers are reset. |
| Supported\_test\_pattern\_modes | 0x200 | 0x0003000F | The pattern generator unit provides a set of known bit sequences or patterns that can be output over the trace port and be detected by the TPA or other associated trace capture device. |
| Current\_test\_pattern\_mode | 0x204 | 0x00000000 | This register indicates the current test pattern or mode selected. Only one of the modes can be set, using bits 17-16, but a multiple number of bits for the patterns can be set using bits 3-0. If timed mode is selected, after the allotted number of cycles is reached, the mode automatically switches to off mode. On reset, this register is set to 18'h00000, off mode with no selected patterns. |
| TPRCR | 0x208 | 0x00000000 | This is an 8-bit counter start value that is decremented. A write sets the initial counter value and a read returns the programmed value. On reset this value is set to 0. |
| FFSR | 0x300 | 0x00000000 | This register indicates the current status of the formatter and flush features available in the TPIU. |
| FFCR | 0x304 | 0x00000000 | This register controls the generation of stop, trigger, and flush events. To disable formatting and put the formatter into bypass mode, bits 1 and 0 must be clear. Setting both bits is the same as setting bit 1. All three flush generating conditions can be enabled together. However, if a second or third flush event is generated from another condition then the current flush completes before the next flush is serviced. Flush from FLUSHIN takes priority over flush from trigger, which in turn completes before a manually activated flush. All Trigger indication conditions can be enabled simultaneously although this can cause the appearance of multiple triggers if flush using trigger is also enabled. Both Stop On settings can be enabled, although if flush on trigger is set up then none of the flushed data is stored. When the system stops, it returns ATREADYS and does not store the accepted data packets. This is to avoid stalling of any other devices that are connected to a trace replicator. If an event in the Formatter and Flush Control Register is required, it must be enabled before the originating event starts. Because requests from flushes and triggers can originate in an asynchronous clock domain, the exact time the component acts on the request cannot be determined with respect to configuring the control. To perform a stop on flush completion through a manually generated flush request, two write operations to the register are required:  • one to enable the stop event, if it is not already enabled  • one to generate the manual flush. |
| FSCR | 0x308 | 0x00000040 | The Formatter Synchronization Counter Register enables effective use on different sized TPAs without wasting large amounts of the storage capacity of the capture device. This counter is the number of formatter frames since the last synchronization packet of 128 bits, and is a 12-bit counter with a maximum count value of 4096. This equates to synchronization every 65536 bytes, that is, 4096 packets x 16 bytes per packet. The default is set up for a synchronization packet every 1024 bytes, that is, every 64 formatter frames. If the formatter is configured for continuous mode, full and half-word sync frames are inserted during normal operation. Under these circumstances, the count value represents the maximum number of complete frames between full synchronization packets. |
| EXTCTL\_In\_Port | 0x400 | 0x00000000 | Two ports can be used as a control and feedback mechanism for any serializers, pin sharing multiplexers or other solutions that might be added to the trace output pins either for pin control or a high speed trace port solution. These ports are raw register banks that sample or export the corresponding external pins. The output register bank is set to all zeros on reset. The input registers sample the incoming signals and as such are Undefined. |
| EXTCTL\_Out\_Port | 0x404 | 0x00000000 | Two ports can be used as a control and feedback mechanism for any serializers, pin sharing multiplexers or other solutions that might be added to the trace output pins either for pin control or a high speed trace port solution. These ports are raw register banks that sample or export the corresponding external pins. The output register bank is set to all zeros on reset. The input registers sample the incoming signals and as such are Undefined. |
| ITTRFLINACK | 0xEE4 | 0x00000000 | The Integration Test Trigger In and Flush In Acknowledge Register enables control of the TRIGINACK and FLUSHINACK outputs from the TPIU. |
| ITTRFLIN | 0xEE8 | 0x00000000 | The Integration Test Trigger In and Flush In Register contains the values of the FLUSHIN and TRIGIN inputs to the TPIU. |
| ITATBDATA0 | 0xEEC | 0x00000000 | The Integration Test ATB Data Register 0 contains the value of the ATDATAS inputs to the TPIU. The values are only valid when ATVALIDS is HIGH. |
| ITATBCTR2 | 0xEF0 | 0x00000000 | The Integration Test ATB Control Register 2 enables control of the ATREADYS and AFVALIDS outputs of the TPIU. |
| ITATBCTR1 | 0xEF4 | 0x00000000 | The Integration Test ATB Control Register 1 contains the value of the ATIDS input to the TPIU. This is only valid when ATVALIDS is HIGH. |
| ITATBCTR0 | 0xEF8 | 0x00000000 | The Integration Test ATB Control Register 0 captures the values of the ATVALIDS, AFREADYS, and ATBYTESS inputs to the TPIU. To ensure the integration registers work correctly in a system, the value of ATBYTESS is only valid when ATVALIDS, bit [0], is HIGH. |
| ITCTRL | 0xF00 | 0x00000000 | This register enables topology detection. This register enables the component to switch from a functional mode, the default behavior, to integration mode where the inputs and outputs of the component can be directly controlled for the purposes of integration testing and topology detection.  After performing integration or topology detection, you must reset the system to ensure correct behavior of CoreSight and other connected system components that the integration or topology detection can affect.  The registers in the TPIU enable the system to set the FLUSHINACK and TRIGINACK output pins. The FLUSHIN and TRIGIN inputs to the TPIU can also be read. The other Integration Test Registers are for testing the integration of the ATB slave interface on the TPIU. |
| CLAIMSET | 0xFA0 | 0x0000000F | Software can use the claim tag to coordinate application and debugger access to trace unit functionality. The claim tags have no effect on the operation of the component. The CLAIMSET Register sets bits in the claim tag, and determines the number of claim bits implemented. |
| CLAIMCLR | 0xFA4 | 0x00000000 | Software can use the claim tag to coordinate application and debugger access to trace unit functionality. The claim tags have no effect on the operation of the component. The CLAIMCLR Register clears the bits in the claim tag, and determines the current value of the claim tag. |
| LAR | 0xFB0 | 0x00000000 | Controls write access from self-hosted, on-chip accesses. The LAR Register does not affect the accesses using the external debugger interface. |
| LSR | 0xFB4 | 0x00000003 | Indicates the status of the lock control mechanism. This lock prevents accidental writes by code being debugged. When locked, write accesses are denied for all registers except for the LAR Register. The lock registers do not affect accesses from the external debug interface. This register reads as 0 when accessed from the external debug interface. |
| AUTHSTATUS | 0xFB8 | 0x00000000 | Reports the required security level and present status. |
| DEVID | 0xFC8 | 0x000000A0 | Indicates the capabilities of the component. |
| DEVTYPE | 0xFCC | 0x00000011 | Provides a debugger with information about the component when the Part Number field is not recognized. The debugger can then report this information. |
| PIDR4 | 0xFD0 | 0x00000004 | Part of the set of peripheral identification registers. Contains part of the designer identity and the memory size. |
| PIDR5 | 0xFD4 | 0x00000000 | Reserved |
| PIDR6 | 0xFD8 | 0x00000000 | Reserved |
| PIDR7 | 0xFDC | 0x00000000 | Reserved |
| PIDR0 | 0xFE0 | 0x00000012 | Part of the set of peripheral identification registers. Contains part of the designer-specific part number. |
| PIDR1 | 0xFE4 | 0x000000B9 | Part of the set of peripheral identification registers. Contains part of the designer-specific part number and part of the designer identity. |
| PIDR2 | 0xFE8 | 0x0000004B | Part of the set of peripheral identification registers. Contains part of the designer identity and the product revision. |
| PIDR3 | 0xFEC | 0x00000000 | Part of the set of peripheral identification registers. Contains the REVAND and CMOD fields. |
| CIDR0 | 0xFF0 | 0x0000000D | A component identification register that indicates the identification registers are present. |
| CIDR1 | 0xFF4 | 0x00000090 | A component identification register that indicates the identification registers are present. This register also indicates the component class. |
| CIDR2 | 0xFF8 | 0x00000005 | A Component Identification Register that indicates that the identification registers are present. |
| CIDR3 | 0xFFC | 0x000000B1 | A component identification register that indicates the identification registers are present. |

* + 1. TMC Register Summary

Table 156. TMC Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| RSZ | 0x004 | 0x00000080 | RAM Size Register |
| STS | 0x00C | 0x0000000C | Status Register |
| RRD | 0x010 |  | RAM Read Data Register trigger. |
| RRP | 0x014 | 0x00000000 | RAM Read Pointer Register external connected trace capture or storage devices. |
| RWP | 0x018 | 0x00000000 | RAM Write Pointer Register |
| TRG | 0x01C | 0x00000000 | Trigger Counter Register |
| CTL | 0x020 | 0x00000000 | Control Register |
| RWD | 0x024 | 0x00000000 | RAM Write Data Register |
| MODE | 0x028 | 0x00000000 | Mode Register |
| LBUFLEVEL | 0x02C | 0x00000000 | Latched Buffer Fill Level |
| CBUFLEVEL | 0x030 | 0x00000040 | Current Buffer Fill Level |
| BUFWM | 0x034 | 0x00000000 | Buffer Level Water Mark |
| RRPHI | 0x038 | 0x00000000 | RAM Read Pointer High Register |
| RWPHI | 0x03C | 0x00000000 | RAM Write Pointer High Register |
| AXICTL | 0x110 | 0x00000000 | AXI Control Register. |
| DBALO | 0x118 | 0x00000000 | Data Buffer Address Low Register |
| DBAHI | 0x11C | 0x00000000 | Data Buffer Address High Register |
| FFSR | 0x300 | 0x00000002 | Formatter and Flush Status Register |
| FFCR | 0x304 | 0x00000000 | Formatter and Flush Control Register |
| PSCR | 0x308 | 0x00000000 | Periodic Synchronization Counter Register |
| ITATBMDATA0 | 0xED0 |  | Integration Test ATB Master Data Register 0 |
| ITATBMCTR2 | 0xED4 | 0x00000000 | Integration Test ATB Master Interface Control 2 Register |
| ITATBMCTR1 | 0xED8 | 0x00000000 | Integration Test ATB Master Control Register 1 |
| ITATBMCTR0 | 0xEDC | 0x00000003 | Integration Test ATB Master Interface Control 0 Register |
| ITMISCOP0 | 0xEE0 | 0x00000000 | Integration Test Miscellaneous Output Register 0 |
| ITTRFLIN | 0xEE8 | 0x000000A0 | Integration Test Trigger In and Flush In Register |
| I TATBDATA0 | 0xFCC | 0x00000011 | Integration Test ATB Data Register 0 |
| ITATBCTR2 | 0xEF0 | 0x00000004 | Integration Test ATB Control 2 Register |
| ITATBCTR1 | 0xEF4 | 0x00000000 | Integration Test ATB Control 1 Register |
| ITATBCTR0 | 0xEF8 | 0x00000000 | Integration Test ATB Control 0 Register |
| ITCTRL | 0xF00 | 0x00000000 | Integration Mode Control Register |
| CLAIMSET | 0xFA0 | 0x0000000F | Claim Tag Set Register |
| CLAIMCLR | 0xFA4 | 0x00000000 | Claim Tag Clear Register |
| LAR | 0xFB0 |  | Lock Access Register |
| LSR | 0xFB4 | 0x00000003 | Lock Status Register |
| AUTHSTATUS | 0xFB8 | 0x0000000D | Authentication Status Register |
| DEVID | 0xFC8 |  | Device Configuration Register |
| DEVTYPE | 0xFCC |  | Device Type Identifier Register |
| PERIPHID4 | 0xFD0 | 0x00000004 | Peripheral ID4 Register |
| PERIPHID5 | 0xFD4 | 0x00000000 | Peripheral ID5 Register |
| PERIPHID6 | 0xFD8 | 0x00000000 | Peripheral ID6 Register |
| PERIPHID7 | 0xFDC | 0x00000000 | Peripheral ID7 Register |
| PERIPHID0 | 0xFE0 | 0x00000061 | Peripheral ID0 Register |
| PERIPHID1 | 0xFE4 | 0x000000B9 | Peripheral ID1 Register |
| PERIPHID2 | 0xFE8 | 0x0000001B | Peripheral ID2 Register |
| PERIPHID3 | 0xFEC | 0x00000000 | Peripheral ID3 Register |
| COMPID0 | 0xFF0 | 0x0000000D | Component ID0 Register |
| COMPID1 | 0xFF4 | 0x00000090 | Component ID1 Register |
| COMPID2 | 0xFF8 | 0x00000005 | Component ID2 Register |
| COMPID3 | 0xFFC | 0x000000B1 | Component ID3 Register |

* + 1. TimeStamp Register Summary

Table 157. TPIU Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| CNTCR | 0x000 | 0x00000000 | Counter Control Register, |
| CNTSR | 0x004 | 0x00000000 | Counter Status Register |
| CNTCVL | 0x008 | 0x00000000 | Current Counter Value Lower register. |
| CNTCVU | 0x00C | 0x00000000 | Current Counter Value Upper register |
| CNTFID0 | 0x020 | 0x00000000 | Base Frequency ID register |
| PIDR4 | 0xFD0 | 0x00000004 | Peripheral ID4 Register |
| PIDR5 | 0xFD4 | 0x00000000 | Peripheral ID5 Register |
| PIDR6 | 0xFD8 | 0x00000000 | Peripheral ID6 Register |
| PIDR7 | 0xFDC | 0x00000000 | Peripheral ID7 Register |
| PIDR0 | 0xFE0 | 0x00000061 | Peripheral ID0 Register |
| PIDR1 | 0xFE4 | 0x000000B9 | Peripheral ID1 Register |
| PIDR2 | 0xFE8 | 0x0000001B | Peripheral ID2 Register |
| PIDR3 | 0xFEC | 0x00000000 | Peripheral ID3 Register |
| CIDR0 | 0xFF0 | 0x0000000D | Component ID0 Register |
| CIDR1 | 0xFF4 | 0x00000090 | Component ID1 Register |
| CIDR2 | 0xFF8 | 0x00000005 | Component ID2 Register |
| CIDR3 | 0xFFC | 0x000000B1 | Component ID3 Register |

* 1. Register Descriptions
     1. APB Interconnect Register Descriptions

ROM table entry

Address: Operational Base + offset (0x000-0x0FC)

Table 158. ROM table entry register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | RO | 19’h1/19’h2  19’h3/19’h4  19’h20 | Base address for master interface 0. Bit[31] is always zero |
| 11:9 | Reserved | - | Reserved |
| 8:4 | RO | 5’b0 | Indicates the power domain ID of the component. This field is only valid if bit[2] is 0b1,otherwise this field must read as zero. Up to 32 power domains are supported using the values from 0x00 to 0x1F. |
| 3 | Reserved |  | Reserved |
| 2 | RO | 1’b0 | Indicates if there is a power domain ID specified in the ROM Table entry.  0 Indicates that the POWER\_DOMAIN\_ID field of this register is not valid.  1 Indicates that the POWER\_DOMAIN\_ID field of this register is valid. |
| 1 | RO | 1’b1 | Indicates the ROM table entry format, for example, 32-bit or other formats  1 ROM table entry is of 32-bit format. |
| 0 | RO | 1’b1 | Indicates if there is a valid ROM entry at this location.  0 Valid ROM table entry not present at this address location.  1 Valid ROM table entry present at this address location. |

Peripheral ID4 Register

Address: Operational Base + offset (0x0FD0)

Table 159. Peripheral ID4 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | This is a 4-bit value that indicates the total contiguous size of the memory window used by this component in powers of 2 from the standard 4KB. For example:  b0000 Indicates that the device occupies 4KB of memory.  b0001 Indicates that the device occupies 8KB of memory.  b0010 Indicates that the device occupies 16KB of memory.  b0011 Indicates that the device occupies 32KB of memory.  b0100 Indicates that the device occupies 64KB of memory.  b0101 Indicates that the device occupies 128KB of memory.  b0110 Indicates that the device occupies 256KB of memory.  b0111 Indicates that the device occupies 512KB of memory.  b1000 Indicates that the device occupies 1MB of memory.  b1001 Indicates that the device occupies 2MB of memory.  b1010 Indicates that the device occupies 4MB of memory.  b1011 Indicates that the device occupies 8MB of memory.  b1100 Indicates that the device occupies 16MB of memory.  b1101 Indicates that the device occupies 32MB of memory.  b1110 Indicates that the device occupies 64MB of memory.  The possible value is:  b0000 Indicates that the device only occupies 4KB of memory. |
| 3:0 | RO | 4’b0 | This is the JEDEC JEP106 continuation code. This code along with, bits[6:4] of the identity code defined in the PIDR0 Register. and with bits[3:0] of the identity code defined in the PIDR1 Register, gives the designer of the component.  This reflects either targeted [11:8] from the DAP, or a sub-system specific value. |

Peripheral ID5-7 Registers

Address: Operational Base + offset (0xFD4 0xFD8 0xFDC)

Table 160. Peripheral ID5-7 Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |

Peripheral ID0 Register

Address: Operational Base + offset (0xFE0)

Table 161. Peripheral ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h8c | Bits[7:0] of the 12 bit part number of the component. The designer of the component assigns this part number. This reflects either targeted [23:16] from the DAP, or a sub-system identifier. |

Peripheral ID1 Register

Address: Operational Base + offset (0xFE4)

Table 162. Peripheral ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’he | Bits[3:0] of the JEDEC JEP106 identity code. This code, along with bits[6:4] of the identity code defined in the PIDR2 Register and the continuation code defined in the PIDR4 Register, gives the designer of the component.  This reflects either the targeted [4:1] from the DAP, or a sub-system identifier. |
| 3:0 | RO | 4’h8 | Bits[11:8] of the 12 bit part number of the component. The designer of the component assigns this part number.  This reflects either the targeted [27:24] from the DAP, or a sub-system identifier. |

Peripheral ID2 Register

Address: Operational Base + offset (0xFE8)

Table 163. Peripheral ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | An incremental value starting from 0b0000 for the first revision of this component. This increases by 1 for both major and minor revisions and is used to identify the major or minor revisions.  This reflects either the targeted [31:28] from the DAP, or a sub-system identifier. |
| 3 | RO | 1‘b1 | Always set. Indicates if the JEDEC assigned designer ID is used.  1b‘1 Indicates that a JEDEC assigned designer ID is used. |
| 2:0 | RO | 4’b0 | Bits [6:4] of the JEDEC JEP106 identity code. This code, along with bits [3:0] of the identity code defined in the PIDR1 Register and the continuation code defined in the PIDR4 register, gives the designer of the component.  This reflects either the targeted [7:5] from the DAP, or a sub-system identifier. |

Peripheral ID3 Register

Address: Operational Base + offset (0xFEC)

Table 164. Peripheral ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Indicates minor errata fixes specific to the revision of the component being used, for example metal fixes after implementation. In most cases, this field is zero. ARM recommends that the component designers ensure that a metal fix can change this field if required, for example, by driving it from registers that reset to zero.  0b0000 Indicates that there are no errata fixes to this component. |
| 3:0 | RO | 4’b0 | Indicates if the customer has modified the behavior of the component. In most cases, this field is zero. The customer changes this value on modifications to this component.  0b0000 Indicates that the customer has not modified this component. |

Component ID0 Register

Address: Operational Base + offset (0xFF0)

Table 165. Component ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h0d | Contains bits[7:0] of the component identification code.  0x0D Bits[7:0] of the identification code. |

Component ID1 Register

Address: Operational Base + offset (0xFF4)

Table 166. Component ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Class of the component, for example, if the component is a ROM table or a generic CoreSight component. Contains bits[15:12] of the component identification code.  b0001 Indicates that the component is a ROM table. |
| 3:0 | RO | 4’b0 | Contains bits [11:8] of the component identification code.  0b0000 Bits [11:8] of the identification code. |

Component ID2 Register

Address: Operational Base + offset (0xFF8)

Table 167. Component ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b05 | Contains bits [23:16] of the component identification code.  0x05 Bits [23:16] of the identification code |

Component ID3 Register

Address: Operational Base + offset (0xFFC)

Table 168. Component ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’hb1 | Contains bits [31:24] of the component identification code.  0xB1 Bits [31:24] of the identification code. |

* + 1. APB-AP Register Descriptions

APB-AP Control/Status Word Register

Address: Operational Base + offset (0x00)

Table 169. APB-AP Control/Status Word Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RW | 1’b0 | Software access enables.  Drives pdbgswen to enable or disable software access to the Debug APB bus in the APB interconnect. The possible values are:  0 Disable software access.  1 Disable software access.  The reset value is b0. On exit from reset, defaults value is b1 to enable software access. |
| 30:12 | Reserved | - | Reserved, SBZ. |
| 11:8 | RW | 4’b0 | Specifies the mode of operation. The possible values are:  b0000 Normal download or upload model.  b0001-b1111 Reserved, SBZ.  The reset value is b0000. |
| 7 | RO | 1’b0 | Transfer in progress. This field indicates if a transfer is currently in progress on the APB master port. |
| 6 | RO | 1’b0 | Indicates the status of the deviceen input.  • If APB-AP is connected to the Debug APB, a bus connected only to debug and trace components; it must be permanently enabled by tying deviceen HIGH. This ensures that trace components can still be programmed when dbgen is LOW. In practice, the APB-AP  is normally used in this way.  • If APB-AP is connected to a system APB dedicated to the non-secure world, deviceen must be connected to dbgen.  • If APB-AP is connected to a system APB dedicated to the secure world, deviceen must be connected to spiden. |
| 5:4 | RW | 2’b0 | Auto address increment and packing mode on Read or Write data access. Increment occurs in word steps. Does not increment if the transaction completes with an error response or the transaction is aborted.  Auto address incrementing is not performed on accesses to banked data registers 0x10-0x1C. The status of these bits is ignored in this case. The possible values are:  b11 Reserved.  b10 Reserved.  b01 Increment.  b00 Auto increment OFF.  The reset value is b00. |
| 3 | Reserved | - | Reserved, SBZ. |
| 2:0 | RO | 2’b10 | Size of the access to perform. The possible value is:  Fixed at b010, 32 bits.  The reset value is b010. |

APB-AP Transfer Address Register

Address: Operational Base + offset (0x04)

Table 170. APB-AP Transfer Address Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | RW | 30’h0 | Address [31:2] Address [31:2] of the current transfer.  Paddr [31:2]=TAR [31:2] for accesses from Data RW Register at 0x0C.  paddr [31:2]=TAR [31:4]+dapcaddr [3:2] for accesses from Banked Data Registers at 0x10-0x1C and 0x0C. |
| 1:0 | RW | 2’b0 | Set to 2'b00. SBZ/RAZ. |

APB-AP Data Read/Write Register

Address: Operational Base + offset (0x0C)

Table 171. APB-AP Data Read/Write Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’h0 | The possible modes are:  Write mode Data value to write for the current transfer.  Read mode Data value read from the current transfer. |

APB-AP Banked Data Registers, BD0-BD3

Address: Operational Base + offset (0x10 - 0x1C)

Table 172. APB-AP Banked Data Registers, BD0-BD3

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’h0 | If dapcaddr [7:4] = 0x0001, it is accessing APB-AP registers in the range 0x10-0x1C, and the derived paddr [31:0] is:  Write mode Data value to write for the current transfer to external address TAR[31:4] + dapcaddr [3:2] + 2'b00.  Read mode Data value read from the current transfer from external address TAR[31:4] + dapcaddr [3:2] + 2'b00.  Auto address incrementing is not performed on DAP accesses to BD0-BD3. The reset value is 0x00000000. |

Debug APB ROM Address

Address: Operational Base + offset (0xF8)

Table 173. Debug APB ROM Address

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 32’h80000003 | Base address of a ROM table. The ROM provides a look-up table for system  components. Bit[1] is SBO. Set bit[0] to 1 if there are debug components on this bus. For most debug APB systems, this value is 0x80000003 |

APB-AP Identification Register

Address: Operational Base + offset (0xFC)

Table 174. APB-AP Identification Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 4’h05 | Revision. Reset value is 0x14 for APB-AP. |
| 27:24 | RO | 4’h4 | JEDEC bank. 0x4 indicates ARM. |
| 23:17 | RO | 8’h3b | JEDEC code. 0x3B indicates ARM. |
| 16 | RO | 1’b1 | Memory AP. 0x1 indicates a standard register map is used. |
| 15:8 | Reserved | - | Reserved, SBZ. |
| 7:0 | RO | 8’h02 | Identity value. The Reset value is 0x03 for APB-AP. |

* + 1. Debug port Register Descriptions

AP Abort Register

Table 175. AP Abort Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:5 | Reserved | - | Reserved, SBZ. |
| 4 | RW | 1’b0 | Write 1 to this bit to clear the STICKYORUN overrun error flag to 0. |
| 3 | RW | 1’b0 | Write 1 to this bit to clear the WDATAERR write data error flag to 0. |
| 2 | RW | 1’b0 | Write 1 to this bit to clear the STICKYERR sticky error flag to 0. |
| 1 | RW | 1’b0 | Write 1 to this bit to clear the STICKYCMP sticky compare flag to 0. |
| 0 | RW | 1’b0 | Write 1 to this bit to generate a DAP abort. This aborts the current AP transaction. Perform this only if the debugger has received WAIT responses over an extended period. |

Identification Code Register

Table 176. Identification Code Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 4’h6 | Version code JTAG-DP 0x5. SW-DP 0x5. |
| 27:12 | RO | 16’hBA00 | Part Number for the debug port. This value is provided by the designer of the debug port and must not be changed. Current ARM-designed debug ports have the following PARTNO values:  JTAG-DP 0xBA00.  SW-DP 0xBA02. |
| 11:1 | RO | 11’h5bb | JEDEC Manufacturer ID, an 11-bit JEDEC code that identifies the designer of the device. This value must not be changed. |
| 0 | RO | 1’b1 | Always 1. |

Control/Status Register, CTRL/STAT

Table 177. Control/Status Register, CTRL/STAT

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 1’b0 | System power-up acknowledge. |
| 30 | RW | 1’b0 | System power-up request.  After a reset this bit is LOW. |
| 29 | RO | 1’b0 | Debug power-up acknowledge. |
| 28 | RW | 1’b0 | Debug power-up request.  After a reset this bit is LOW. |
| 27 | RO | 1’b0 | Debug reset acknowledge. |
| 26 | RW | 1’b0 | Debug reset request.  After a reset this bit is LOW. |
| 25:24 | Reserved | - | Reserved, RAZ/SBZP. |
| 23:12 | RW | 12’h0 | Transaction counter.  After a reset the value of this field is UNPREDICTABLE. |
| 11:8 | RW | 4’b0 | Indicates the bytes to be masked in pushed compare and pushed verify operations.  After a reset the value of this field is UNPREDICTABLE. |
| 7 | RO | 1’b0 | This bit is set to 1 if a Write Data Error occurs. It is set if:  • there is a parity or framing error on the data phase of a write  • A write that the debug port accepted is then discarded without being submitted to the access port.  This bit can only be cleared by writing b1 to the WDERRCLR field of the Abort Register .After a power-on reset this bit is LOW. |
| 6 | RO | 1’b0 | This bit is set to 1 if the response to a previous access port or RDBUFF was OK. It is cleared to 0 if the response was not OK.  This flag always indicates the response to the last access port read access.  After a power-on reset this bit is LOW. |
| 5 | RO | 1’b0 | This bit is set to 1 if an error is returned by an access port transaction. To clear this bit:  JTAG-DP Write b1 to this bit of this register.  SW-DP Write b1 to the STKERRCLR field of the Abort Register.  After a power-on reset this bit is LOW. |
| 4 | RO | 1’b0 | This bit is set to 1 when a match occurs on a pushed compare or a pushed verify operation. To clear this bit:  JTAG-DP Write b1 to this bit of this register.  SW-DP Write b1 to the STKCMPCLR field of the Abort Register.  After a power-on reset this bit is LOW. |
| 3:2 | RW | 2’b0 | This field sets the transfer mode for access port operations.  After a power-on reset the value of this field is UNPREDICTABLE. |
| 1 | RO | 1’b0 | If overrun detection is enabled, this bit is set to 1 when an overrun occurs. To clear this bit:  JTAG-DP Write b1 to this bit of this register.  SW-DP Write b1 to the ORUNERRCLR field of the Abort Register.  After a power-on reset this bit is LOW. See bit [0] of this register. |
| 0 | RW | 1‘b0 | This bit is set to b1 to enable overrun detection.  After a reset this bit is LOW. |

AP Select Register

Table 178. AP Select Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RO | 8’b0 | Selects the current access port. The possible values are:  0x00 Selects the AP connected to master interface 0 of the DAPBUS interconnect.  0x01 Selects the AP connected to master interface 1 of the DAPBUS interconnect, if present.  0x02 Selects the AP connected to master interface 2 of the DAPBUS interconnect, if present.  0x03 Selects the AP connected to master interface 3 of the DAPBUS interconnect, if present.  0x1F Selects the AP connected to master interface 31 of the DAPBUS interconnect, if present.  The reset value of this field is UNPREDICTABLE. |
| 23:8 | Reserved | - | Reserved. |
| 7:4 | RO | 4’b0 | Selects the active 4-word register window on the current access port.  The reset value of this field is UNPREDICTABLE. |
| 3:0 | RW | 4’b0 | Selects the register that appears at DP register 0x4. The possible values are:  0x0 CTRL/STAT, RW.  0x1 DLCR, RW.  0x2 TARGETID, RO.  0x3 DLPIDR, RO.  All other values are reserved. Writing a reserved value to this field is UNPREDICTABLE. |

Data Link Control Register

Table 179. Data Link Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | Reserved | - | Reserved |
| 9:8 | RW | 2’b0 | Turnaround state period |
| 7:6 | RO | 2’b1 | Identifies the operating mode for the wire connection to the debug port |
| 5:3 | Reserved | 3’b0 | Reserved |
| 2:0 | Reserved | 3’b0 | Reserved |

Data Link Protocol Identification Register

Table 180. Data Link Protocol Identification Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | RO | 4’b0 | Configuration-dependent  This field defines a unique instance number for this device within the system. This value must be unique for all devices that are connected together in a multi-drop system with identical values in the  TREVISION fields in the TARGETID Register. The value of this field reflects the value of the instanceid [3:0]input. |
| 27:4 | Reserved. | - | Reserved. |
| 3:0 | RO | 4’b0 | Defines the serial wire protocol version. This value is 0x1, which indicates SW protocol version 2 |

* + 1. CTI Register Descriptions

CTI Control Register

Address: Operational Base + offset (0x000)

Table 181. CTI Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 0 | RW. | 1’b0 | Enables or disables the ECT. The possible values are:  0 When this bit is 0, all cross triggering mapping logic functionality is disabled.  1 When this bit is 1, cross triggering mapping logic functionality is enabled. |

CTI Interrupt Acknowledge Register

Address: Operational Base + offset (0x010)

Table 182. CTI Interrupt Acknowledge Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | WO. | 8’b0 | Acknowledges the corresponding CTITRIGOUT output. There is one bit of the register for each CTITRIGOUT output. When a 1 is written to a bit in this register, the corresponding CTITRIGOUT is acknowledged and is cleared when MAPTRIGOUT is LOW. Writing a 0 to any of the bits in this register has no effect. |

CTI Application Trigger Set Register

Address: Operational Base + offset (0x014)

Table 183. CTI Application Trigger Set Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW. | 4’b0 | Setting a bit HIGH generates a channel event for the selected channel. There is one bit of the register for each channel.  Reads as follows:  0 Application trigger is inactive.  1 Application trigger is active.  Writes as follows:  0 No effect.  1 Generate channel event. |

CTI Application Trigger Clear Register

Address: Operational Base + offset (0x018)

Table 184. CTI Application Trigger Clear Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | WO. | 4’b0 | Clears corresponding bits in the CTIAPPSET Register. There is one bit of the register for each channel. When a 1 is written to a bit in this register, the corresponding application trigger is disabled in the CTIAPPSET Register. Writing a 0 to any of the bits in this register has no effect. |

CTI Application Pulse Register

Address: Operational Base + offset (0x01C)

Table 185. CTI Application Pulse Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | WO. | 4’b0 | Setting a bit HIGH generates a channel event pulse for the selected channel. There is one bit of the register for each channel. When a 1 is written to a bit in this register, a corresponding channel event pulse is generated for one CTICLK period. Writing a 0 to any of the bits in this register has no effect. |

CTI Trigger 0 to Channel Enable Register

Address: Operational Base + offset (0x020)

Table 186. CTI Trigger 0 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 1 to Channel Enable Register

Address: Operational Base + offset (0x024)

Table 187. CTI Trigger 1 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when a CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 2 to Channel Enable Register

Address: Operational Base + offset (0x028)

Table 188. CTI Trigger 2 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 3 to Channel Enable Register

Address: Operational Base + offset (0x02C)

Table 189. CTI Trigger 3 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 4 to Channel Enable Register

Address: Operational Base + offset (0x030)

Table 190. CTI Trigger 4 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when a CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 5 to Channel Enable Register

Address: Operational Base + offset (0x034)

Table 191. CTI Trigger 5 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 6 to Channel Enable Register

Address: Operational Base + offset (0x038)

Table 192. CTI Trigger 6 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Trigger 7 to Channel Enable Register

Address: Operational Base + offset (0x03C)

Table 193. CTI Trigger 7 to Channel Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, it enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. For example, TRIGINEN [0] set to 1 enables CTITRIGIN onto channel 0. Writing a 0 to any of the bits in this register disables the CTITRIGIN signal from generating an event on the respective channel of the CTM. Reading this register returns the programmed value. |

CTI Channel to Trigger 0 Enable Register

Address: Operational Base + offset (0x0A0)

Table 194. CTI Channel to Trigger 0 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [1] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT[0] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 1 Enable Register

Address: Operational Base + offset (0x0A4)

Table 195. CTI Channel to Trigger 1 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [1] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [1] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 2 Enable Register

Address: Operational Base + offset (0x0A8)

Table 196. CTI Channel to Trigger 2 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [2] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [2] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 3 Enable Register

Address: Operational Base + offset (0x0AC)

Table 197. CTI Channel to Trigger 3 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [3] output. There is one bit of the field for each of the four channels. When  A 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [3] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 4 Enable Register

Address: Operational Base + offset (0x0B0)

Table 198. CTI Channel to Trigger 4 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [4] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [4] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 5 Enable Register

Address: Operational Base + offset (0x0B4)

Table 199. CTI Channel to Trigger 5 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [5] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [5] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 6 Enable Register

Address: Operational Base + offset (0x0B8)

Table 200. CTI Channel to Trigger 6 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [6] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [6] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Channel to Trigger 7 Enable Register

Address: Operational Base + offset (0x0BC)

Table 201. CTI Channel to Trigger 7 Enable Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate a CTITRIGOUT [7] output. There is one bit of the field for each of the four channels. When a 1 is written to a bit in this register, the channel input, CTICHIN, from the CTM is routed to the CTITRIGOUT output. For example, enabling bit 0 enables CTICHIN [0] to cause a trigger event on the CTITRIGOUT [7] output. When a 0 is written to any of the bits in this register, the channel input, CTICHIN, from the CTM is not routed to the CTITRIGOUT output. Reading this register returns the programmed value. |

CTI Trigger In Status Register

Address: Operational Base + offset (0x130)

Table 202. CTI Trigger In Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b0 | Shows the status of the CTITRIGIN inputs. There is one bit of the field for each trigger input. The possible values are:  1 CTITRIGIN is active.  0 CTITRIGIN is inactive.  Because the register provides a view of the raw CTITRIGIN inputs, the reset value is UNKNOWN. |

CTI Trigger Out Status Register

Address: Operational Base + offset (0x134)

Table 203. CTI Trigger Out Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b0 | Shows the status of the CTITRIGOUT outputs. There is one bit of the field for each trigger output. The possible values are:  1 CTITRIGOUT is active.  0 CTITRIGOUT is inactive. |

CTI Channel In Status Register

Address: Operational Base + offset (0x138)

Table 204. CTI Channel In Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RO | 4’b0 | Shows the status of the CTICHIN inputs. There is one bit of the field for each channel input. The possible values are:  0 CTICHIN is inactive.  1 CTICHIN is active.  Because the register provides a view of the raw CTICHIN inputs, the reset value is UNKNOWN. |

CTI Channel Out Status Register

Address: Operational Base + offset (0x13C)

Table 205. CTI Channel Out Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RO | 4’b0 | Shows the status of the CTICHOUT outputs. There is one bit of the field for each channel output. The possible values are:  0 CTICHOUT is inactive.  1 CTICHOUT is active. |

Enable CTI Channel Gate Register

Address: Operational Base + offset (0x140)

Table 206. Enable CTI Channel Gate Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3 | RW | 1’b1 | Enable CTICHOUT3. Set to 0 to disable channel propagation. |
| 2 | RW | 1’b1 | Enable CTICHOUT2. Set to 0 to disable channel propagation |
| 1 | RW | 1’b1 | Enable CTICHOUT1. Set to 0 to disable channel propagation. |
| 0 | RW | 1’b1 | Enable CTICHOUT0. Set to 0 to disable channel propagation. |

External Multiplexer Control Register

Address: Operational Base + offset (0x144)

Table 207. External Multiplexer Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b0 | IMPLEMENTATION DEFINED ASIC control, value written to the register is output on ASICCTL [7:0]. If external multiplexing of trigger signals is implemented then the number of multiplexed signals on each trigger must be reflected within the Device ID Register. This is done within a Verilog define EXTMUXNUM. |

ITCHINACK Register

Address: Operational Base + offset (0xEDC)

Table 208. ITCHINACK Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | WO | 4’b0 | Set the value of the CTCHINACK outputs. |

ITCHOUT Register

Address: Operational Base + offset (0xEE4)

Table 209. ITCHOUT Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | WO | 4’b0 | Set the value of the CTCHOUT outputs. |

ITTRIGOUT Register

Address: Operational Base + offset (0xEE8)

Table 210. ITTRIGOUT Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | WO | 8’b0 | Set the value of the CTTRIGOUT outputs. |

ITCHOUTACK Register

Address: Operational Base + offset (0xEEC)

Table 211. ITCHOUTACK Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RO | 4’b0 | Read the values of the CTCHOUTACK inputs. |

ITCHIN Register

Address: Operational Base + offset (0xEF4)

Table 212. ITCHIN Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RO | 4’b0 | Read the value of the CTCHIN inputs. |

ITTRIGIN Register

Address: Operational Base + offset (0xEF8)

Table 213. ITTRIGIN Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b0 | Read the values of the CTTRIGIN inputs. |

Integration Mode Control Register

Address: Operational Base + offset (0xF00)

Table 214. Integration Mode Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 0 | RW | 1’b1 | Enables the component to switch from functional mode to integration mode and back. If no integration functionality is implemented, this register must read as zero.  0 Disable integration modes.  1 Enable integration mode. |

Claim Tag Set Register

Address: Operational Base + offset (0xFA0)

Table 215. Claim Tag Set Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’hf | On reads, for each bit:  1 Claim tag bit is implemented  On writes, for each bit:  0 Has no effect.  1 Sets the relevant bit of the claim tag |

Claim Tag Clear Register

Address: Operational Base + offset (0xFA4)

Table 216. Claim Tag Clear Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | On reads, for each bit:  0 Claim tag bit is not set.  1 Claim tag bit is set.  On writes, for each bit:  0 Has no effect.  1 Clears the relevant bit of the claim tag. |

Lock Access Register

Address: Operational Base + offset (0xFB0)

Table 217. Lock Access Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 32’b0 | When you write 0xC5ACCE55, subsequent write operations to this device are enabled. Any other value disables subsequent write operations. |

Lock Status Register

Address: Operational Base + offset (0xFB4)

Table 218. Lock Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:3 | Reserved | - | Reserved |
| 2 | RO | 1’b0 | Indicates that the LAR Register is implemented as 32-bit.  0b0 This component implements a 32-bit LAR Register. |
| 1 | RO | 1’b1 | Returns the present lock status of the device.  0b0 Indicates that write operations are permitted in this device.  0b1 Indicates that write operations are not permitted in this device. Read operations are permitted. |
| 0 | RO | 1’b1 | Indicates that a lock control mechanism is present in this device.  0b0 Indicates that a lock control mechanism is not present in this device. Write operations to the LAR Register are ignored.  0b1 Indicates that a lock control mechanism is present in this device. |

Authentication Status Register

Address: Operational Base + offset (0xFB8)

Table 219. Authentication Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:6 | RO | 2’b0 | Indicates the security level for secure non-invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 5:4 | RO | 2’b0 | Indicates the security level for secure invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 3:2 | RO | 2’b1 | Indicates the security level for non-secure non-invasive debug:  0b10 Functionality is disabled.  0b11 Functionality is enabled. |
| 1:0 | RO | 2’b1 | Indicates the security level for non-secure invasive debug:  0b10 Functionality is disabled.  0b11 Functionality is enabled. |

Device Configuration Register

Address: Operational Base + offset (0xFC8)

Table 220. Device Configuration Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:20 | Reserved | - | Reserved |
| 19:16 | RO | 4’h4 | Number of ECT channels available. |
| 15:8 | RO | 8’h08 | Number of ECT triggers available. |
| 7:5 | Reserved | - | Reserved |
| 4:0 | RO | 5’b0 | Indicates the number of multiplexing available on Trigger Inputs and Trigger Outputs using ASICCTL. The default value of 5'b00000 indicates that no multiplexing is present. Reflects the value of the Verilog define EXTMUXNUM that you must alter accordingly. |

Device Type Identifier Register

Address: Operational Base + offset (0xFCC)

Table 221. Device Type Identifier Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’h1 | Sub-classification of the type of the debug component as specified in the CoreSight Architecture Specification within the major classification as specified in the MAJOR field.  b0001 Indicates that this component is a cross-triggering component. |
| 3:0 | RO | 4’h4 | Major classification of the type of the debug component as specified in the CoreSight Architecture Specification for this debugs and trace component.  0b0100 Indicates that this component allows a debugger to control other components in a CoreSight system. |

Peripheral ID4 Register

Address: Operational Base + offset (0xFD0)

Table 222. Peripheral ID4 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4‘b0000 | This is a 4-bit value that indicates the total contiguous size of the memory window used by this component in powers of 2 from the standard 4KB. For example:  b0000 Indicates that the device occupies 4KB of memory.  b0001 Indicates that the device occupies 8KB of memory.  b0010 Indicates that the device occupies 16KB of memory.  b0011 Indicates that the device occupies 32KB of memory.  b0100 Indicates that the device occupies 64KB of memory.  b0101 Indicates that the device occupies 128KB of memory.  b0110 Indicates that the device occupies 256KB of memory.  b0111 Indicates that the device occupies 512KB of memory.  b1000 Indicates that the device occupies 1MB of memory.  b1001 Indicates that the device occupies 2MB of memory.  b1010 Indicates that the device occupies 4MB of memory.  b1011 Indicates that the device occupies 8MB of memory.  b1100 Indicates that the device occupies 16MB of memory.  b1101 Indicates that the device occupies 32MB of memory.  b1110 Indicates that the device occupies 64MB of memory.  The possible value is:  b0000 Indicates that the device only occupies 4KB of memory. |
| 3:0 | RO | 4‘b0100 | This is the JEDEC JEP106 continuation code. This code, along with bits [6:4] of the identity code defined in the PIDR0 Register and bits[3:0] of the identity code defined in the PIDR1 Register, gives the designer of the component.  0b0100 JEDEC continuation code. |

Peripheral ID5-7 Registers

Address: Operational Base + offset (0xFD4 0xFD8 0xFDC)

Table 223. Peripheral ID5-7 Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |

Peripheral ID0 Register

Address: Operational Base + offset (0xFE0)

Table 224. Peripheral ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h8c | Bits[7:0] of the 12 bit part number of the component. The designer of the component assigns this part number. This reflects either targeted[23:16] from the DAP, or a sub-system identifier. |

Peripheral ID1 Register

Address: Operational Base + offset (0XFE4)

Table 225. Peripheral ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | - | Reserved |
| 7:4 | RO | 4’he | Bits[3:0] of the JEDEC JEP106 identity code. This code, along with bits[6:4] of the identity code defined in the PIDR2 Register and the continuation code defined in the PIDR4 Register, gives the designer of the component.  This reflects either the targeted [4:1] from the DAP, or a sub-system identifier. |
| 3:0 | RO | 4’h8 | Bits[11:8] of the 12 bit part number of the component. The designer of the component assigns this part number.  This reflects either the targeted [27:24] from the DAP, or a sub-system identifier. |

Peripheral ID2 Register

Address: Operational Base + offset (0xFE8)

Table 226. Peripheral ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | An incremental value starting from 0b0000 for the first revision of this component. This increases by 1 for both major and minor revisions and is used to identify the major or minor revisions.  This reflects either the targeted [31:28] from the DAP, or a sub-system identifier. |
| 3 | RO | 1‘b1 | Always set. Indicates if the JEDEC assigned designer ID is used.  0b1 Indicates that a JEDEC assigned designer ID is used. |
| 2:0 | RO | 4’b0 | Bits[6:4] of the JEDEC JEP106 identity code. This code, along with bits[3:0] of the identity code defined in the PIDR1 Register and the continuation code defined in the PIDR4 register, gives the designer of the component.  This reflects either the targeted [7:5] from the DAP, or a sub-system identifier. |

Peripheral ID3 Register

Address: Operational Base + offset (0xFEC)

Table 227. Peripheral ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Indicates minor errata fixes specific to the revision of the component being used, for example metal fixes after implementation. In most cases, this field is zero. ARM recommends that the component designers ensure that a metal fix can change this field if required, for example, by driving it from registers that reset to zero.  0b0000 Indicates that there are no errata fixes to this component. |
| 3:0 | RO | 4’b0 | Indicates if the customer has modified the behavior of the component. In most cases, this field is zero. The customer changes this value on modifications to this component.  0b0000 Indicates that the customer has not modified this component. |

Component ID0 Register

Address: Operational Base + offset (0xFF0)

Table 228. Component ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h0d | Contains bits[7:0] of the component identification code.  0x0D Bits[7:0] of the identification code. |

Component ID1 Register

Address: Operational Base + offset (0xFF4)

Table 229. Component ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Class of the component, for example, if the component is a ROM table or a generic CoreSight component. Contains bits[15:12] of the component identification code.  b0001 Indicates that the component is a ROM table. |
| 3:0 | RO | 4’b0 | Contains bits [11:8] of the component identification code.  0b0000 Bits [11:8] of the identification code. |

Component ID2 Register

Address: Operational Base + offset (0xFF8)

Table 230. Component ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved |  | Reserved |
| 7:0 | RO | 8’b05 | Contains bits [23:16] of the component identification code.  0x05 Bits[23:16] of the identification code |

Component ID3 Register

Address: Operational Base + offset (0xFFC)

Table 231. Component ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’hb1 | Contains bits [31:24] of the component identification code.  0xB1 Bits [31:24] of the identification code. |

* + 1. TPIU Register Descriptions

Supported Port Size Register

Address: Operational Base + offset (0x000)

Table 232. Supported Port Size Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 1’b0 | Indicates whether the TPIU supports port size of 32 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 30 | RO | 1’b0 | Indicates whether the TPIU supports port size of 31 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 29 | RO | 1’b0 | Indicates whether the TPIU supports port size of 30 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 28 | RO | 1’b0 | Indicates whether the TPIU supports port size of 29 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 27 | RO | 1’b0 | Indicates whether the TPIU supports port size of 28 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 26 | RO | 1’b0 | Indicates whether the TPIU supports port size of 27 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 25 | RO | 1’b0 | Indicates whether the TPIU supports port size of 26 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 24 | RO | 1’b0 | Indicates whether the TPIU supports port size of 25 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 23 | RO | 1’b0 | Indicates whether the TPIU supports port size of 24 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 22 | RO | 1’b0 | Indicates whether the TPIU supports port size of 23 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 21 | RO | 1’b0 | Indicates whether the TPIU supports port size of 22 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 20 | RO | 1’b0 | Indicates whether the TPIU supports port size of 21 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 19 | RO | 1’b0 | Indicates whether the TPIU supports port size of 20 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 18 | RO | 1’b0 | Indicates whether the TPIU supports port size of 19 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 17 | RO | 1’b0 | Indicates whether the TPIU supports port size of 18 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 16 | RO | 1’b0 | Indicates whether the TPIU supports port size of 17 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 15 | RO | 1’b0 | Indicates whether the TPIU supports port size of 16 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 14 | RO | 1’b0 | Indicates whether the TPIU supports port size of 15 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 13 | RO | 1’b0 | Indicates whether the TPIU supports port size of 14 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 12 | RO | 1’b0 | Indicates whether the TPIU supports port size of 13 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 11 | RO | 1’b0 | Indicates whether the TPIU supports port size of 12 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported |
| 10 | RO | 1’b0 | Indicates whether the TPIU supports port size of 11 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 9 | RO | 1’b0 | Indicates whether the TPIU supports port size of 10 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 8 | RO | 1’b0 | Indicates whether the TPIU supports port size of 9 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 7 | RO | 1’b0 | Indicates whether the TPIU supports port size of 8 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 6 | RO | 1’b0 | Indicates whether the TPIU supports port size of 7 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 5 | RO | 1’b0 | Indicates whether the TPIU supports port size of 6 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 4 | RO | 1’b0 | Indicates whether the TPIU supports port size of 5 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 3 | RO | 1’b0 | Indicates whether the TPIU supports port size of 4 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 2 | RO | 1’b0 | Indicates whether the TPIU supports port size of 3 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 1 | RO | 1’b0 | Indicates whether the TPIU supports port size of 2 bits. The possible values are:  0 Port size is not supported.  1 Port size is supported. |
| 0 | RO | 1’b1 | Indicates whether the TPIU supports port size of 1 bit. The possible values are:  0 Port size is not supported.  1 Port size is supported. |

Current Port Size Register

Address: Operational Base + offset (0x004)

Table 233. Current Port Size Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 32 bits. The possible values are:  0 Current Port size is not 32.  1 Current Port size is 32. |
| 30 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 31 bits. The possible values are:  0 Current Port size is not 31.  1 Current Port size is 31. |
| 29 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 30 bits. The possible values are:  0 Current Port size is not 30.  1 Current Port size is 30. |
| 28 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 29 bits. The possible values are:  0 Current Port size is not 29.  1 Current Port size is 29. |
| 27 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 28 bits. The possible values are:  0 Current Port size is not 28.  1 Current Port size is 28. |
| 26 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 27 bits. The possible values are:  0 Current Port size is not 27.  1 Current Port size is 27. |
| 25 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 26 bits. The possible values are:  0 Current Port size is not 26.  1 Current Port size is 26. |
| 24 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 25 bits. The possible values are:  0 Current Port size is not 25.  1 Current Port size is 25. |
| 23 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 24 bits. The possible values are:  0 Current Port size is not 24.  1 Current Port size is 24. |
| 22 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 23 bits. The possible values are:  0 Current Port size is not 23.  1 Current Port size is 23. |
| 21 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 22 bits. The possible values are:  0 Current Port size is not 22.  1 Current Port size is 22. |
| 20 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 21 bits. The possible values are:  0 Current Port size is not 21.  1 Current Port size is 21. |
| 19 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 20 bits. The possible values are:  0 Current Port size is not 20.  1 Current Port size is 20. |
| 18 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 19 bits. The possible values are:  0 Current Port size is not 19.  1 Current Port size is 19. |
| 17 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 18 bits. The possible values are:  0 Current Port size is not 18.  1 Current Port size is 18. |
| 16 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 17 bits. The possible values are:  0 Current Port size is not 17.  1 Current Port size is 17. |
| 15 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 16 bits. The possible values are:  0 Current Port size is not 16.  1 Current Port size is 16. |
| 14 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 15 bits. The possible values are:  0 Current Port size is not 15.  1 Current Port size is 15. |
| 13 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 14 bits. The possible values are:  0 Current Port size is not 14.  1 Current Port size is 14. |
| 12 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 13 bits. The possible values are:  0 Current Port size is not 13.  1 Current Port size is 13. |
| 11 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 12 bits. The possible values are:  0 Current Port size is not 12.  1 Current Port size is 12. |
| 10 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 11 bits. The possible values are:  0 Current Port size is not 11.  1 Current Port size is 11. |
| 9 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 10 bits. The possible values are:  0 Current Port size is not 10.  1 Current Port size is 10. |
| 8 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 9 bits. The possible values are:  0 Current Port size is not 9.  1 Current Port size is 9. |
| 7 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 8 bits. The possible values are:  0 Current Port size is not 8.  1 Current Port size is 8. |
| 6 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 7 bits. The possible values are:  0 Current Port size is not 7.  1 Current Port size is 7. |
| 5 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 6 bits. The possible values are:  0 Current Port size is not 6.  1 Current Port size is 6. |
| 4 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 5 bits. The possible values are:  0 Current Port size is not 5.  1 Current Port size is 5. |
| 3 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 4 bits. The possible values are:  0 Current Port size is not 4.  1 Current Port size is 4. |
| 2 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 3 bits. The possible values are:  0 Current Port size is not 3.  1 Current Port size is 3. |
| 1 | RW | 1’b0 | Indicates whether the current port size of the TPIU is 2 bits. The possible values are:  0 Current Port size is not 2.  1 Current Port size is 2. |
| 0 | RW | 1’b1 | Indicates whether the current port size of the TPIU is 1 bit. The possible values are:  0 Current Port size is not 1.  1 Current Port size is 1. |

Supported Trigger Modes Register

Address: Operational Base + offset (0x100)

Table 234. Supported Trigger Modes Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:18 | Reserved | - | Reserved |
| 17 | RO | 1’b0 | A trigger has occurred but the counter is not at zero. The possible values are:  0 Either a trigger has not occurred or the counter is at zero.  1 A trigger has occurred but the counter is not at zero. |
| 16 | RO | 1’b0 | A trigger has occurred and the counter has reached zero. The possible values are:  0 Trigger not occurred.  1 Trigger occurred. |
| 15:9 | Reserved | - | Reserved |
| 8 | RO | 1’b1 | Indicates whether an 8-bit wide counter register implemented. The possible values are:  0 8-bit wide counter register not implemented.  1 8-bit wide counter register implemented. |
| 7:5 | Reserved | - | Reserved |
| 4 | RO | 1’b1 | Indicates whether multiply the Trigger Counter by 65536 is supported. The possible values are:  0 Multiply the Trigger Counter by 65536 not supported.  1 Multiply the Trigger Counter by 65536 supported. |
| 3 | RO | 1’b1 | Indicates whether multiply the Trigger Counter by 256 is supported. The possible values are:  0 Multiply the Trigger Counter by 256 not supported.  1 Multiply the Trigger Counter by 256 supported. |
| 2 | RO | 1’b1 | Indicates whether multiply the Trigger Counter by 16 is supported. The possible values are:  0 Multiply the Trigger Counter by 16 not supported.  1 Multiply the Trigger Counter by 16 supported. |
| 1 | RO | 1’b1 | Indicates whether multiply the Trigger Counter by 4 is supported. The possible values are:  0 Multiply the Trigger Counter by 4 not supported.  1 Multiply the Trigger Counter by 4 supported. |
| [0 | RO | 1’b1 | Indicates whether multiply the Trigger Counter by 2 is supported. The possible values are:  0 Multiply the Trigger Counter by 2 not supported.  1 Multiply the Trigger Counter by 2 supported. |

Trigger Counter Value Register

Address: Operational Base + offset (0x104)

Table 235. Trigger Counter Value Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RW | 8’h00 | 8-bit counter value for the number of words to be output from the formatter before a trigger is inserted. At reset the value is zero and this value has the effect of disabling the register, that is, there is no delay. |

Trigger Multiplier Register

Address: Operational Base + offset (0x108)

Table 236. Trigger Multiplier Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:5 | Reserved |  | Reserved |
| 4 | RO | 1’b0 | Multiply the Trigger Counter by 65536 (2^16). The possible values are:  0 Multiplier disabled.  1 Multiplier enabled. |
| 3 | RW | 1’b0 | Multiply the Trigger Counter by 256 (2^8). The possible values are:  0 Multiplier disabled.  1 Multiplier enabled. |
| 2 | RW | 1’b0 | Multiply the Trigger Counter by 16 (2^4). The possible values are:  0 Multiplier disabled.  1 Multiplier enabled. |
| 1 | RW | 1’b0 | Multiply the Trigger Counter by 4 (2^2). The possible values are:  0 Multiplier disabled.  1 Multiplier enabled. |
| 0 | RW | 1’b0 | Multiply the Trigger Counter by 2 (2^1). The possible values are:  0 Multiplier disabled.  1 Multiplier enabled. |

Supported Test Patterns/Modes Register

Address: Operational Base + offset (0x200)

Table 237. Supported Test Patterns/Modes Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:18 | Reserved | - | Reserved |
| 17 | RO | 1’b1 | Indicates whether continuous mode is supported. The possible values are:.  1 Mode supported. |
| 16 | RO | 1’b1 | Indicates whether timed mode is supported. The possible values are:  1 Mode supported. |
| 15:4 | Reserved | - | Reserved |
| 3 | RO | 1’b1 | FF/00 pattern supported to be output over the trace port. The possible values are:  1 Pattern supported. |
| 2 | RO | 1’b1 | AA/55 pattern supported to be output over the trace port. The possible values are:  1 Pattern supported. |
| 1 | RO | 1’b1 | Walking 0s Pattern supported to be output over the trace port. The possible values are:  1 Pattern supported. |
| 0 | RO | 1’b1 | Walking 1s Pattern supported to be output over the trace port. The possible values are:  1 Pattern supported. |

Current Test Pattern/Modes Register

Address: Operational Base + offset (0x204)

Table 238. Current Test Pattern/Modes Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:18 | Reserved | - | Reserved |
| 17 | RW | 1’b0 | Indicates whether continuous mode is supported. The possible values are:  0 Mode not supported.  1 Mode supported. |
| 16 | RW | 1’b0 | Indicates whether timed mode is supported. The possible values are:  0 Mode not supported.  1 Mode supported. |
| 15:4 | Reserved | - | Reserved |
| 3 | RW | 1’b0 | FF/00 pattern supported to be output over the trace port. The possible values are:  0 Pattern not supported.  1 Pattern supported. |
| 2 | RW | 1’b0 | AA/55 pattern supported to be output over the trace port. The possible values are:  0 Pattern not supported.  1 Pattern supported. |
| 1 | RW | 1’b0 | Walking 0s Pattern supported to be output over the trace port. The possible values are:  0 Pattern not supported.  1 Pattern supported. |
| 0 | RW | 1’b0 | Walking 1s Pattern supported to be output over the trace port. The possible values are:  0 Pattern not supported.  1 Pattern supported. |

TPIU Test Pattern Repeat Counter Register

Address: Operational Base + offset (0x208)

Table 239. TPIU Test Pattern Repeat Counter Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RW | 8’h0 | 8-bit counter value to indicate the number of TRACECLKIN cycles that a pattern runs for before switching to the next pattern. The default value is 0. |

Formatter and Flush Status Register

Address: Operational Base + offset (0x300)

Table 240. Formatter and Flush Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:3 | Reserved | - | Reserved |
| 2 | RO | 1’b0 | Indicates whether the TRACECTL pin is available for use. If this bit is set then TRACECTL is present. If no TRACECTL pin is available, that is, this bit is zero, then the data formatter must be used and only in continuous mode. This is constrained by the CSTPIU\_TRACECTL\_VAL Verilog define, that is not user-modifiable, and the external tie-off TPCTL. If either constraint reports LOW, then no TRACECTL is present and this inability to use the pin is reflected in this register. The possible values are:  0 TRACECTL pin not present.  1 TRACECTL pin present. |
| 1 | RO | 1’b0 | The formatter has received a stop request signal and all trace data and post-amble is sent. Any more trace data on the ATB interface is ignored and ATREADYS goes HIGH. The possible values are:  0 Formatter has not stopped.  1 Formatter has stopped. |
| 0 | RO | 1’b0 | This is an indication of the current state of AFVALIDS. The possible values are:  0 AFVALIDS is LOW.  1 AFVALIDS is HIGH. |

Formatter and Flush Control Register

Address: Operational Base + offset (0x304)

Table 241. Formatter and Flush Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:14 | Reserved | - | Reserved |
| 13 | RW | 1’b0 | Stop the formatter after a trigger event is observed. Reset to disabled or zero. The possible values are:  0 Disable stopping the formatter after a trigger event is observed.  1 Enable stopping the formatter after a trigger event is observed. |
| 12 | RW | 1’b0 | This forces the FIFO to drain off any part-completed packets. Setting this bit enables this function but this is clear on reset, or disabled. The possible values are:  0 Disable stopping the formatter on return of AFREADYS.  1 Enable stopping the formatter on return of AFREADYS. |
| 11 | Reserved | - | Reserved |
| 10 | RW | 1’b0 | Indicates a trigger on flush completion on AFREADYS being returned. The possible values are:  0 Disable trigger indication on return of AFREADYS.  1 Enable trigger indication on return of AFREADYS. |
| 9 | RW | 1’b0 | Indicate a trigger on a trigger event. The possible values are:  0 Disable trigger indication on a trigger event.  1 Enable trigger indication on a trigger event. |
| 8 | RW | 1’b0 | Indicate a trigger on TRIGIN being asserted. The possible values are:  0 Disable trigger indication when TRIGIN is asserted.  1 Enable trigger indication when TRIGIN is asserted. |
| 7 | Reserved | - | Reserved |
| 6 | RW | 1’b0 | Setting this bit causes a flush to be generated. This is cleared when this flush is serviced. This bit is clear on reset. The possible values are:  0 Manual flush is not initiated.  1 Manual flush is initiated. |
| 5 | RW | 1’b0 | Set this bit to initiates a manual flush of data in the system when a trigger event occurs. On reset this bit is clear. A trigger event is defined as when the trigger counter reaches zero, or in the case of the trigger counter being zero, when TRIGIN is HIGH. The possible values are:  0 Disable generation of flush when a Trigger Event occurs.  1 Enable generation of flush when a Trigger Event occurs. |
| 4 | RW | 1’b0 | Set this bit to enable use of the FLUSHIN connection. This is clear on reset. The possible values are:  0 Disable generation of flush using the FLUSHIN interface.  1 Enable generation of flush using the FLUSHIN interface. |
| 3:2 | Reserved | - | Reserved |
| 1 | RW | 1’b0 | Embed in trigger packets and indicate null cycles using sync packets. Reset value is this bit clear. Can only be changed when FtStopped is HIGH. The possible values are:  0 Continuous formatting disabled.  1 Continuous formatting enabled. |
| 0 | RW | 1’b0 | Do not embed triggers into the formatted stream. Trace disable cycles and triggers are indicated by TRACECTL, where fitted. On reset this bit clear. Can only be changed when FtStopped is HIGH. The possible values are:  0 Formatting disabled.  1 Formatting enabled. |

Formatter Synchronization Counter Register

Address: Operational Base + offset (0x308)

Table 242. Formatter Synchronization Counter Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | Reserved | - | Reserved |
| 11:0 | RW | 12’h040 | 12-bit counter value to indicate the number of complete frames between full synchronization packets. The default value is 64, 0x40. |

TPIU EXCTL Port Register IN

Address: Operational Base + offset (0x400)

Table 243. TPIU EXCTL Port Register IN

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h0 | EXTCTL inputs. |

TPIU EXCTL Port Register Out

Address: Operational Base + offset (0x404)

Table 244. TPIU EXCTL Port Register Out

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved |  | Reserved |
| 7:0 | RO | 8’h0 | EXTCTL outputs. |

Integration Test Trigger In and Flush In Acknowledge Register

Address: Operational Base + offset (0xEE4)

Table 245. Integration Test Trigger In and Flush In Acknowledge Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | WO | 1’b0 | Set the value of FLUSHINACK. The possible values are:  0 Set the value of FLUSHINACK to 0.  1 Set the value of FLUSHINACK to 1. |
| 0 | WO | 1’b0 | Set the value of TRIGINACK. The possible values are:  0 Set the value of TRIGINACK to 0.  1 Set the value of TRIGINACK to 1. |

Integration Test Trigger In and Flush In Register

Address: Operational Base + offset (0xEE8)

Table 246. Integration Test Trigger In and Flush In Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | RO | 1’b0 | Read the value of FLUSHIN. The possible values are:  0 FLUSHIN is LOW.  1 FLUSHIN is HIGH. |
| 0 | RO | 1’b0 | Read the value of TRIGIN. The possible values are:  0 TRIGIN is LOW.  1 TRIGIN is HIGH. |

Integration Test ATB Data Register 0

Address: Operational Base + offset (0xEEC)

Table 247. Integration Test ATB Data Register 0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:5 | Reserved | - | Reserved |
| 4 | RO | 1’b0 | Read the value of ATDATAS [31]. The possible values are:  0b1 ATDATAS [31] is 1.  0b0 ATDATAS [31] is 0. |
| 3 | RO | 1’b0 | Read the value of ATDATAS [23]. The possible values are:  0b1 ATDATAS [23] is 1.  0b0 ATDATAS [23] is 0. |
| 2 | RO | 1’b0 | Read the value of ATDATAS [15]. The possible values are:  0b1 ATDATAS [15] is 1.  0b0 ATDATAS [15] is 0. |
| 1 | RO | 1’b0 | Read the value of ATDATAS [7]. The possible values are:  0b1 ATDATAS [7] is 1.  0b0 ATDATAS [7] is 0. |
| 0 | RO | 1’b0 | Read the value of ATDATAS [0]. The possible values are:  0b1 ATDATAS [0] is 1.  0b0 ATDATAS [0] is 0. |

Integration Test ATB Control Register 2

Address: Operational Base + offset (0xEF0)

Table 248. Integration Test ATB Control Register 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | WO | 1’b0 | Set the value of AFVALID. The possible values are:  0 Set the value of AFVALID to 0.  1 Set the value of AFVALID to 1. |
| 0 | WO | 1’b0 | Set the value of ATREADY. The possible values are:  0 Set the value of ATREADY to 0.  1 Set the value of ATREADY to 1. |

Integration Test ATB Control Register 1

Address: Operational Base + offset (0xEF4)

Table 249. Integration Test ATB Control Register 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:7 | Reserved | - | Reserved |
| 6:0 | RO | 7’h0 | Read the value of ATIDS. |

Integration Test ATB Control Register 0

Address: Operational Base + offset (0xEF8)

Table 250. Integration Test ATB Control Register 0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | Reserved | - | Reserved |
| 9:8 | RO | 1’b0 | Read the value of ATBYTESS. |
| 7:2 | Reserved | - | Reserved |
| 1 | RO | 1’b0 | Read the value of AFREADYS. The possible values are:  0 AFREADYS is 0.  1 AFREADYS is 1. |
| 0 | RO | 1’b0 | Read the value of ATVALIDS. The possible values are:  0 ATVALIDS is 0.  1 ATVALIDS is 1. |

Integration Mode Control Register

Address: Operational Base + offset (0xF00)

Table 251. Integration Mode Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 0 | RW | 1’b0 | Enables the component to switch from functional mode to integration mode and back. If no integration functionality is implemented, this register must read as zero.  0 Disable integration mode.  1 Enable integration mode. |

Claim Tag Set Register

Address: Operational Base + offset (0xFA0)

Table 252. Claim Tag Set Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 3:0 | RW | 4’hF | On reads, for each bit:  1 Claim tag bit is implemented  On writes, for each bit:  0 Has no effect.  1 Sets the relevant bit of the claim tag |

Claim Tag Clear Register

Address: Operational Base + offset (0xFA4)

Table 253. Claim Tag Clear Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 3:0 | RO | 8’hb1 | On reads, for each bit:  0 Claim tag bit is not set.  1 Claim tag bit is set.  On writes, for each bit:  0 Has no effect.  1 Clears the relevant bit of the claim tag. |

Lock Access Register

Address: Operational Base + offset (0xFB0)

Table 254. Lock Access Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’b0 | When you write 0xC5ACCE55, subsequent write operations to this device are enabled. Any other value disables subsequent write operations. |

Lock Status Register

Address: Operational Base + offset (0xFB4)

Table 255. Lock Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:3 | Reserved | - | Reserved |
| 2 | WO | 1’b0 | Indicates that the LAR Register is implemented as 32-bit.  0b0 This component implements a 32-bit LAR Register. |
| 1 | WO | 1’b0 | Returns the present lock status of the device.  0b0 Indicates that write operations are permitted in this device.  0b1 Indicates that write operations are not permitted in this device. Read operations are permitted. |
| 0 | WO | 1’b0 | Indicates that a lock control mechanism is present in this device.  0b0 Indicates that a lock control mechanism is not present in this device. Write operations to the LAR Register are ignored.  0b1 Indicates that a lock control mechanism is present in this device. |

Authentication Status Register

Address: Operational Base + offset (0xFB8)

Table 256. Authentication Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:6 | RO | 2’b0 | Indicates the security level for secure non-invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 5:4 | RO | 2’b0 | Indicates the security level for secure invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 3:2 | RO | 2’b0 | Indicates the security level for non-secure non-invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 1:0 | RO | 2’h3 | Indicates the security level for non-secure invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |

Device Configuration Register

Address: Operational Base + offset (0xFC8)

Table 257. Device Configuration Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | Reserved | - | Reserved |
| 11 | RO | 1’b0 | Indicates if Serial Wire Output, UART or NRZ, is supported. The possible value is:  0 Serial Wire Output, UART or NRZ, is not supported. |
| 10 | RO | 1’b0 | Indicates if Serial Wire Output, Manchester encoded format, is supported.  0 Serial Wire Output, Manchester encoded format, is not supported. |
| 9 | RO | 1’b0 | Indicates if trace clock plus data is supported. The possible value is:  0 Trace clock and data is supported. |
| 8:6 | RO | 3’b010 | FIFO size in powers of 2. The possible value is:  0b010 FIFO size of 4 entries, that is, 16 bytes. |
| 5 | RO | 1’b1 | Indicates the relationship between ATCLK and TRACECLKIN. The possible value is:  1 ATCLK and TRACECLKIN are asynchronous. |
| 4:0 | RO | 5’b0 | Indicates the hidden level of input multiplexing. When non-zero this value indicates the type or number of ATB multiplexing present on the input to the ATB. Currently only 0x00 is supported, that is, no multiplexing is present. This value is used to assist in topology detection of the ATB structure. |

Device Type Identifier Register

Address: Operational Base + offset (0xFCC)

Table 258. Device Type Identifier Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Sub-classification of the type of the debug component as specified in the CoreSight Architecture Specification within the major classification as specified in the MAJOR field.  b0001 Indicates that this component is a trace port component. |
| 3:0 | RO | 4’h3 | Major classification of the type of the debug component as specified in the CoreSight Architecture Specification for this debugs and trace component.  b0001 Indicates that this component is a trace sink component. |

Peripheral ID4 Register

Address: Operational Base + offset (0xFD0)

Table 259. Peripheral ID4 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’h1 | This is a 4-bit value that indicates the total contiguous size of the memory window used by this component in powers of 2 from the standard 4KB. For example:  b0000 Indicates that the device occupies 4KB of memory.  b0001 Indicates that the device occupies 8KB of memory.  b0010 Indicates that the device occupies 16KB of memory.  b0011 Indicates that the device occupies 32KB of memory.  b0100 Indicates that the device occupies 64KB of memory.  b0101 Indicates that the device occupies 128KB of memory.  b0110 Indicates that the device occupies 256KB of memory.  b0111 Indicates that the device occupies 512KB of memory.  b1000 Indicates that the device occupies 1MB of memory.  b1001 Indicates that the device occupies 2MB of memory.  b1010 Indicates that the device occupies 4MB of memory.  b1011 Indicates that the device occupies 8MB of memory.  b1100 Indicates that the device occupies 16MB of memory.  b1101 Indicates that the device occupies 32MB of memory.  b1110 Indicates that the device occupies 64MB of memory.  The possible value is:  b0000 Indicates that the device only occupies 4KB of memory. |
| 3:0 | RO | 4’h2 | This is the JEDEC JEP106 continuation code. This code, along with bits [6:4] of the identity code defined in the PIDR0 Register and bits [3:0] of the identity code defined in the PIDR1 Register, gives the designer of the component.  0b0100 JEDEC continuation code. |

Peripheral ID5-7 Registers

Address: Operational Base + offset (0xFD4 0xFD8 0xFDC)

Table 260. Peripheral ID5-7 Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |

Peripheral ID0 Register

Address: Operational Base + offset (0xFE0)

Table 261. Peripheral ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h8c | Bits[7:0] of the 12 bit part number of the component. The designer of the component assigns this part number. This reflects either targeted [23:16] from the DAP, or a sub-system identifier. |

Peripheral ID1 Register

Address: Operational Base + offset (0XFE4)

Table 262. Peripheral ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | - | Reserved |
| 7:4 | RO | 4’he | Bits[3:0] of the JEDEC JEP106 identity code. This code, along with bits[6:4] of the identity code defined in the PIDR2 Register and the continuation code defined in the PIDR4 Register, gives the designer of the component.  This reflects either the targeted [4:1] from the DAP, or a sub-system identifier. |
| 3:0 | RO | 4’h8 | Bits[11:8] of the 12 bit part number of the component. The designer of the component assigns this part number.  This reflects either the targeted [27:24] from the DAP, or a sub-system identifier. |

Peripheral ID2 Register

Address: Operational Base + offset (0xFE8)

Table 263. Peripheral ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | An incremental value starting from 0b0000 for the first revision of this component. This increases by 1 for both major and minor revisions and is used to identify the major or minor revisions.  This reflects either the targeted [31:28] from the DAP, or a sub-system identifier. |
| 3 | RO | 1;b1 | Always set. Indicates if the JEDEC assigned designer ID is used.  0b1 Indicates that a JEDEC assigned designer ID is used. |
| 2:0 | RO | 4’b0 | Bits[6:4] of the JEDEC JEP106 identity code. This code, along with bits[3:0] of the identity code defined in the PIDR1 Register and the continuation code defined in the PIDR4 register, gives the designer of the component.  This reflects either the targeted [7:5] from the DAP, or a sub-system identifier. |

Peripheral ID3 Register

Address: Operational Base + offset (0xFEC)

Table 264. Peripheral ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Indicates minor errata fixes specific to the revision of the component being used, for example metal fixes after implementation. In most cases, this field is zero. ARM recommends that the component designers ensure that a metal fix can change this field if required, for example, by driving it from registers that reset to zero.  0b0000 Indicates that there are no errata fixes to this component. |
| 3:0 | RO | 4’b0 | Indicates if the customer has modified the behavior of the component. In most cases, this field is zero. The customer changes this value on modifications to this component.  0b0000 Indicates that the customer has not modified this component. |

Component ID0 Register

Address: Operational Base + offset (0xFF0)

Table 265. Component ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h0d | Contains bits[7:0] of the component identification code.  0x0D Bits[7:0] of the identification code. |

Component ID1 Register

Address: Operational Base + offset (0xFF4)

Table 266. Component ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Class of the component, for example, if the component is a ROM table or a generic CoreSight component. Contains bits[15:12] of the component identification code.  b0001 Indicates that the component is a ROM table. |
| 3:0 | RO | 4’b0 | Contains bits[11:8] of the component identification code.  0b0000 Bits[11:8] of the identification code. |

Component ID2 Register

Address: Operational Base + offset (0xFF8)

Table 267. Component ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b05 | Contains bits[23:16] of the component identification code.  0x05 Bits[23:16] of the identification code |

Component ID3 Register

Address: Operational Base + offset (0xFFC)

Table 268. Component ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’hb1 | Contains bits[31:24] of the component identification code.  0xb1 Bits[31:24] of the identification code. |

* + 1. TMC Register Descriptions

TMC RAM Depth Register

Address: Operational Base + offset (0x004)

Table 269. TMC RAM Depth Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 32’h 00000080 | Defines the depth, in words, of the trace RAM. |

TMC Status Register

Address: Operational Base + offset (0x00C)

Table 270. TMC Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3 | RO | 1 | Formatter pipeline is empty. All data is stored to RAM. The possible values are:  0 Formatter pipeline is not empty.  1 Formatter pipeline is empty. |
| 2 | RO | 1 | The acquisition complete flag indicates that the capture is completed when the formatter stops because of any of the methods defined in the Formatter and Flush Control Register, or TraceCaptEn = 0. This also results in the FtStopped bit in the Formatter and Flush Status Register going HIGH. The possible values are:  0 Acquisition is not complete.  1 Acquisition is complete. |
| 1 | RO | 0 | The Triggered bit is set when the component observes a trigger. This does not indicate that the formatter embedded a trigger in the trace data, but is determined when programming the Formatter bit and Flush Control Registers. The possible values are:  0 A trigger is not observed.  1 A trigger is observed. |
| 0 | RO | 0 | The flag indicates if the RAM is full or not. The possible values are:  0 RAM write pointer is not wrapped around and so the RAM is not full.  1 RAM write pointer is wrapped around and so the RAM is full. |

TMC RAM Read Data Register

Address: Operational Base + offset (0x010)

Table 271. TMC RAM Read Data Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 32’b0 | Data read from the ETB Trace RAM |

TMC RAM Read Pointer Register

Address: Operational Base + offset (0x014)

Table 272. TMC RAM Read Pointer Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | Reserved | - | Reserved |
| 9:0 | RW | 10’b0 | Sets the read pointer to the required value. The read pointer is used to read entries from the Trace RAM through the APB interface. |

TMC RAM Write Pointer Register

Address: Operational Base + offset (0x018)

Table 273. TMC RAM Write Pointer Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | Reserved | - | Reserved |
| 9:0 | RW | 10’b0 | The RAM Write Pointer Register sets the write pointer to the required value. The Write Pointer is used to write entries from the CoreSight bus to the Trace RAM. |

TMC Trigger Counter Register

Address: Operational Base + offset (0x01C)

Table 274. TMC Trigger Counter Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | Reserved | - | Reserved |
| 9:0 | RW | 10’b0 | The counter is used as follows:  Trace after The counter is set to a large value, slightly less than the number of entries in the RAM.  Trace before The counter is set to a small value.  Trace about The counter is set to half the depth of the trace RAM.  You must not write to this register when trace capture is enabled, FtStopped=0 and TraceCaptEn=1. If a write is attempted, then the register is not updated. A read operation is permitted when trace capture is enabled. |

TMC Control Register

Address: Operational Base + offset (0x020)

Table 275. TMC Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 0 | RW | 32’b0 | ETB Trace Capture Enable. This is the master enable bit forcing FtStopped HIGH when TraceCaptEn is LOW. When capture is disabled, any remaining data in the ATB formatter is stored to RAM. When all of the data are stored, the formatter outputs FtStopped. Capture is fully disabled, or complete, when FtStopped goes HIGH.  The possible values are:  0 Disable trace capture.  1 Enable trace capture. |

TMC RAM Write Data Register

Address: Operational Base + offset (0x024)

Table 276. TMC RAM Write Data Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 32’b0 | When the trace capture is disabled and writes the data to the ETB trace RAM, the contents of this register are placed into the ETB Trace RAM.  Writing to these register increments the RAM Write Pointer Register value. If trace capture is enabled, and this register is accessed, then a read from this register outputs 0xFFFFFFFF.  Reads of this register never increment the RAM Write Pointer Register. A constant stream of 1s being output corresponds to a synchronization output from the ETB.  If a write access is attempted, the data is not written into trace RAM. |

Latched Buffer Fill Level Register

Address: Operational Base + offset (0x02C)

Table 277. Latched Buffer Fill Level Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 32’b0 | Indicates the maximum fill level of the trace memory in 32-bit words since this register was last read |

Current Buffer Fill Level Register

Address: Operational Base + offset (0x030)

Table 278. Current Buffer Fill Level Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 32’b0 | Indicates the current fill level of the trace memory in 32-bit words |

BUFWM Register

Address: Operational Base + offset (0x034)

Table 279. BUFWM Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 32’b0 | Indicates the required threshold vacancy level in 32-bit words in the trace memory |

RRPHI Register

Address: Operational Base + offset (0x038)

Table 280. RRPHI Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RW | 8’b0 | Bits [39:32] of the read pointer |

RWPHI Register

Address: Operational Base + offset (0x03C)

Table 281. RWPHI Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RW | 8’b0 | Bits[39:32] of the write pointer |

AXICTL Register

Address: Operational Base + offset (0x110)

Table 282. AXICTL Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | Reserved | - | Reserved |
| 11:8 | RW | 4’b0 | This field indicates the maximum number of data transfers that can occur within each burst initiated by the TMC on the AXI master interface. The Write burst initiated on the AXI can be of a smaller length that the programmed value in the case when the formatter has stopped because of a stop Condition having occurred. Programming this field to a burst length value greater than the write buffer depth results in a burst length that is equal to the write buffer depth. The burst length programmed must be compatible with the trace buffer size and the AXI data width so that the total number of bytes of data transferred in a burst is not greater than the trace buffer size or, if scatter-gather operation is enabled, is not greater than 4KB. Programming an incompatible burst length results in Unpredictable behavior. It is recommended that this value be set to no more than half the write buffer depth. Also, it is recommended that this value be set to enable an AXI burst of at least one frame of trace data.This field is decoded as follows:  0x0 One data transfer per burst. This is the default.  0x1 Maximum of two data transfers per burst.  ...  0xF Maximum of 16 data transfers per burst. |
| 7 | RW | 1’b0 | This bit indicates whether trace memory is accessed as a single buffer in system memory or as a  Linked-list based scatter-gather memory. This bit is ignored when in Disabled state.  0 Trace memory is a single contiguous block of system memory.  1 Trace memory is spread over multiple blocks of system memory based on a  Linked-list mechanism. For more information |
| 6 | RW | 1’b0 | Reserved. |
| 5 | RW | 1’b0 | This bit controls the value driven on the ARCACHEM[3]or AWCACHEM[3]signal on the AXI master interface when performing AXI transfers. If CacheCtrlBit1 is LOW, this bit must also be  LOW to comply with the AXI protocol. Setting this bit to HIGH when the CacheCtrlBit1 is LOW  Results in Unpredictable behavior.  0 Do not cache allocate on writes.  1 Cache allocate on writes. |
| 4 | RW | 1’b0 | This bit controls the value driven on the ARCACHEM[2]or AWCACHEM[2]signal on the AXI master interface when performing AXI transfers. If CacheCtrlBit1 is LOW, this bit must also be LOW to comply with the AXI protocol. Setting this bit to HIGH when the CacheCtrlBit1 is LOW  Results in Unpredictable behavior.  0 Do not cache allocate on reads.  1 Cache allocate on reads. |
| 3 | RW | 1’b0 | This bit controls the value driven on the ARCACHEM[1]or AWCACHEM[1]signal on the AXI master interface when performing AXI transfers.  0 Non-cacheable.  1 Cacheable. |
| 2 | RW | 1’b0 | This bit controls the value driven on the ARCACHEM[0]or AWCACHEM[0]signal on the AXI master interface when performing AXI transfers.  0 Non-bufferable.  1 Bufferable. |
| 1 | RW | 1’b0 | This bit controls the value driven on ARPROTM[1]or AWPROTM[ 1 ]on the AXI master interface when performing AXI transfers.  0 Secure access.  1 Non-secure access. |
| 0 | RW | 1’b0 | This bit controls the value driven on ARPROTM[0]or AWPROTM[ 0 ]on the AXI master interface when performing AXI transfers.  0 Normal access.  1 Privileged access. A privileged processing mode might have a greater level of access within a system. |

DBALO Register

Address: Operational Base + offset (0x118)

Table 283. DBALO Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’b0 | Holds the lower 32 bits of the 40-bit address used to locate the trace buffer in system memory |

DBAHI Register

Address: Operational Base + offset (0x11C)

Table 284. DBAHI Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RW | 8’b0 | Holds the upper eight bits of the 40-bit address used to locate the trace buffer in system memory |

TMC Formatter and Flush Status Register

Address: Operational Base + offset (0x300)

Table 285. TMC Formatter and Flush Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | RO | 1’b1 | Formatter stopped. The formatter has received a stop request signal and all trace data and post-amble is sent. Any more trace data on the ATB interface is ignored and ATREADYS goes HIGH. The possible values are:  0 Formatter is not stopped.  1 Formatter is stopped. |
| 0 | RO | 1’b0 | Flush In Progress. This is an indication of the current state of AFVALIDS. The possible values are:  0 AFVALIDS is LOW.  1 AFVALIDS is HIGH. |

TMC Formatter and Flush Control Register

Address: Operational Base + offset (0x304)

Table 286. TMC Formatter and Flush Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:14 | Reserved | - | Reserved |
| 13 | RW | 1’b0 | Stop the formatter after a Trigger Event is observed. Reset to disabled, that is, zero. The possible values are:  0 Disable stopping the formatter after a Trigger Event is observed.  1 Enable stopping the formatter after a Trigger Event is observed. |
| 12 | RW | 1’b0 | This forces the FIFO to drain off any part-completed packets. Setting this bit enables this function but this is clear on reset, that is, disabled. The possible values are:  0 Disable stopping the formatter on return of AFREADYS.  1 Enable stopping the formatter on return of AFREADYS. |
| 11 | Reserved | - | Reserved |
| 10 | RW | 1’b0 | Indicates a trigger on Flush completion, AFREADYS is returned. The possible values are:  0 Disable trigger indication when AFREADYS is returned.  1 Enable trigger indication when AFREADYS is returned. |
| 9 | RW | 1’b0 | Indicate a trigger on a Trigger Event. The possible values are:  0 Disable trigger indication on a Trigger event.  1 Enable trigger indication on a Trigger event. |
| 8 | RW | 1’b0 | Indicate a trigger when TRIGIN is asserted. The possible values are:  0 Disable trigger indication when TRIGIN is asserted.  1 Enable trigger indication when TRIGIN is asserted. |
| 7 | Reserved | - | Reserved |
| 6 | RW | 1’b0 | Setting this bit initiates a manual flush. This is cleared after this flush is serviced. This bit is clear on reset. The possible values are:  0 Manual flush is not initiated.  1 Manual flush is initiated. |
| 5 | RW | 1’b0 | Generate flush using Trigger event. Set this bit to cause a flush of data in the system when a Trigger Event occurs. This bit is clear on reset. A Trigger Event is defined as when the Trigger counter reaches zero, that is, fitted or, in the case of the trigger counter being zero, that is, are not fitted, when TRIGIN is HIGH. The possible values are:  0 Disable generation of flush when a trigger event occurs.  1 Enable generation of flush when a trigger event occurs. |
| 4 | RW | 1’b0 | Set this bit to enable use of the FLUSHIN connection. This is clear on reset. The possible values are:  0 Disable generation of flush using the FLUSHIN interface.  1 Enable generation of flush using the FLUSHIN interface. |
| 3:2 | Reserved | - | Reserved |
| 1 | RW | 1’b0 | Continuous mode in the ETB corresponds to normal mode with the embedding of triggers. Can only be changed when FtStopped is HIGH. This bit is clear on reset. The possible values are:  0 Continuous formatting disabled.  1 Continuous formatting enabled. |
| 0 | RW | 1’b0 | Do not embed Triggers into the formatted stream. Trace disable cycles and triggers are indicated by TRACECTL, where fitted. Can only be changed when FtStopped is HIGH. This bit is clear on reset. The possible values are:  0 Formatting disabled.  1 Formatting enabled. |

Integration Test Miscellaneous Output Register 0

Address: Operational Base + offset (0xEE0)

Table 287. Integration Test Miscellaneous Output Register 0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | RW | 1’b0 | Set the value of FULL. The possible values are:  0 Set the value of FULL to 0.  1 Set the value of FULL to 1. |
| 0 | RW | 1’b0 | Set the value of ACQCOMP. The possible values are:  0 Set the value of ACQCOMP to 0.  1 Set the value of ACQCOMP to 1. |

Integration Test Trigger In and Flush In Acknowledge Register

Address: Operational Base + offset (0xEE4)

Table 288. Integration Test Trigger In and Flush In Acknowledge Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | WO | 1’b0 | Set the value of FLUSHINACK. The possible values are:  0 Set the value of FLUSHINACK to 0.  1 Set the value of FLUSHINACK to 1. |
| 0 | WO | 1’b0 | Set the value of TRIGINACK. The possible values are:  0 Set the value of TRIGINACK to 0.  1 Set the value of TRIGINACK to 1. |

Integration Test Trigger In and Flush In Register

Address: Operational Base + offset (0xEE8)

Table 289. Integration Test Trigger In and Flush In Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | RO | 1’b0 | Read the value of FLUSHIN. The possible values are:  0 FLUSHIN is LOW.  1 FLUSHIN is HIGH. |
| 0 | RO | 1’b0 | Read the value of TRIGIN. The possible values are:  0 TRIGIN is LOW.  1 TRIGIN is HIGH. |

Integration Test ATB Data Register 0

Address: Operational Base + offset (0xEEC)

Table 290. Integration Test ATB Data Register 0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:5 | Reserved | - | Reserved |
| 4 | WO | 1’b0 | Read the value of ATDATAS [31]. The possible values are:  0 ATDATAS [31] is 0.  1 ATDATAS [31] is 1. |
| 3 | WO | 1’b0 | Read the value of ATDATAS [23]. The possible values are:  0 ATDATAS [23] is 0.  1 ATDATAS [23] is 1. |
| 2 | WO | 1’b0 | Read the value of ATDATAS [15]. The possible values are:  0 ATDATAS [15] is 0.  1 ATDATAS [15] is 1. |
| 1 | WO | 1’b0 | Read the value of ATDATAS [7].  0 ATDATAS [7] is 0.  1 ATDATAS [7] is 1. |
| 0 | WO | 1’b0 | Read the value of ATDATAS [0]. The possible values are:  0 ATDATAS [0] is 0.  1 ATDATAS [0] is 1. |

Integration Test ATB Control Register 2

Address: Operational Base + offset (0xEF0)

Table 291. Integration Test ATB Control Register 2

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:2 | Reserved | - | Reserved |
| 1 | WO | 1’b0 | Set the value of AFVALIDS. The possible values are:  0 Set the value of AFVALIDS to 0.  1 Set the value of AFVALIDS to 1. |
| 0 | WO | 1’b0 | Set the value of ATREADYS. The possible values are:  0 Set the value of ATREADYS to 0.  1 Set the value of ATREADYS to 1. |

Integration Test ATB Control Register 1

Address: Operational Base + offset (0xEF4)

Table 292. Integration Test ATB Control Register 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:7 | Reserved | - | Reserved |
| 6:0 | WO | 7’b0 | Read the value of ATIDS. |

Integration Test ATB Control Register 0

Address: Operational Base + offset (0xEF8)

Table 293. Integration Test ATB Control Register 0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | Reserved | - | Reserved |
| 9:8 | WO | 2’b0 | Read the value of ATBYTESS. |
| 7:2 | Reserved | - | Reserved |
| 1 | WO | 1’b0 | Read the value of AFREADYS. The possible values are:  0 AFREADYS is 0.  1 AFREADYS is 1. |
| 0 | WO | 1’b0 | Read the value of ATVALIDS. The possible values are:  0 ATVALIDS is 0.  1 ATVALIDS is 1. |

Integration Mode Control Register

Address: Operational Base + offset (0xF00)

Table 294. Integration Mode Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | Reserved | - | Reserved |
| 0 | WO | 1’b0 | Enables the component to switch from functional mode to integration mode and back. If no integration functionality is implemented, this register must read as zero.  0 Disable integration mode.  1 Enable integration mode. |

Claim Tag Set Register

Address: Operational Base + offset (0xFA0)

Table 295. Claim Tag Set Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | -- | Reserved |
| 3:0 | RW | 4’hF | SET On reads, for each bit:  1 Claim tag bit is implemented.  On writes, for each bit:  0 Has no effect.  1 Sets the relevant bit of the claim tag. |

Claim Tag Clear Register

Address: Operational Base + offset (0xFA4)

Table 296. Claim Tag Clear Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | Reserved | - | Reserved |
| 3:0 | RW | 4’b0 | On reads, for each bit:  0 Claim tag bit is not set.  1 Claim tag bit is set.  On writes, for each bit:  0 Has no effect.  1 Clears the relevant bit of the claim tag. |

Lock Access Register

Address: Operational Base + offset (0xFB0)

Table 297. Lock Access Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 32’b0 | When you write 0xC5ACCE55, subsequent write operations to this device are enabled. Any other value disables subsequent write operations. |

Lock Status Register

Address: Operational Base + offset (0xFB4)

Table 298. Lock Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:3 | Reserved | -- | Reserved |
| 2 | RO | 1’b0 | Indicates that the LAR Register is implemented as 32-bit.  0b0 This component implements a 32-bit LAR Register. |
| 1 | RO | 1’b1 | Returns the present lock status of the device.  0b0 Indicates that write operations are permitted in this device.  0b1 Indicates that write operations are not permitted in this device. Read operations are permitted. |
| 0 | RO | 1’b1 | Indicates that a lock control mechanism is present in this device.  0b0 Indicates that a lock control mechanism is not present in this device. Write operations to the LAR Register are ignored.  0b1 Indicates that a lock control mechanism is present in this device. |

Authentication Status Register

Address: Operational Base + offset (0xFB8)

Table 299. Authentication Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:6 | RO | 2’b0 | Indicates the security level for secure non-invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 5:4 | RO | 2’b0 | Indicates the security level for secure invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 3:2 | RO | 2’b0 | Indicates the security level for non-secure non-invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |
| 1:0 | RO | 2’b0 | Indicates the security level for non-secure invasive debug:  0b00 Functionality is not implemented or is controlled elsewhere. |

Device Configuration Register

Address: Operational Base + offset (0xFC8)

Table 300. Device Configuration Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:6 | Reserved | - | Reserved |
| 5 | RO | 1’b0 | This bit returns 0 on reads to indicate that the ETB RAM operates synchronously to ATCLK.  0 The ETB RAM operates synchronously to ATCLK. |
| 4:0 | RO | 4’b0 | When non-zero, this value indicates the type or number of ATB multiplexing present on the input to the ATB.  b0000 Only 0x00 is supported, that is, no multiplexing is present. This value is used to assist topology detection of the ATB structure. |

Device Type Identifier Register

Address: Operational Base + offset (0xFCC)

Table 301. Device Type Identifier Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Sub-classification of the type of the debug component as specified in the CoreSight Architecture Specification within the major classification as specified in the MAJOR field.  b0010 This component is a trace buffer, ETB. |
| 3:0 | RO | 4’b0 | Major classification of the type of the debug component as specified in the CoreSight Architecture Specification for this debugs and trace component.  b0001 This component is a trace sink component. |

Peripheral ID4 Register

Address: Operational Base + offset (0xFD0)

Table 302. Peripheral ID4 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | This is a 4-bit value that indicates the total contiguous size of the memory window used by this component in powers of 2 from the standard 4KB. For example:  b0000 Indicates that the device occupies 4KB of memory.  b0001 Indicates that the device occupies 8KB of memory.  b0010 Indicates that the device occupies 16KB of memory.  b0011 Indicates that the device occupies 32KB of memory.  b0100 Indicates that the device occupies 64KB of memory.  b0101 Indicates that the device occupies 128KB of memory.  b0110 Indicates that the device occupies 256KB of memory.  b0111 Indicates that the device occupies 512KB of memory.  b1000 Indicates that the device occupies 1MB of memory.  b1001 Indicates that the device occupies 2MB of memory.  b1010 Indicates that the device occupies 4MB of memory.  b1011 Indicates that the device occupies 8MB of memory.  b1100 Indicates that the device occupies 16MB of memory.  b1101 Indicates that the device occupies 32MB of memory.  b1110 Indicates that the device occupies 64MB of memory.  The possible value is:  b0000 Indicates that the device only occupies 4KB of memory. |
| 3:0 | RO | 4’h4 | This is the JEDEC JEP106 continuation code. This code, along with bits[6:4] of the identity code defined in the PIDR0 Register and bits[3:0] of the identity code defined in the PIDR1 Register, gives the designer of the component.  0b0100 JEDEC continuation code. |

Peripheral ID5-7 Registers

Address: Operational Base + offset (0xFD4 0xFD8 0xFDC)

Table 303. Peripheral ID5-7 Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |

Peripheral ID0 Register

Address: Operational Base + offset (0xFE0)

Table 304. Peripheral ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h8c | Bits[7:0] of the 12 bit part number of the component. The designer of the component assigns this part number. This reflects either targeted [23:16] from the DAP, or a sub-system identifier. |

Peripheral ID1 Register

Address: Operational Base + offset (0XFE4)

Table 305. Peripheral ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | - | Reserved |
| 7:4 | RO | 4’he | Bits [3:0] of the JEDEC JEP106 identity code. This code, along with bits [6:4] of the identity code defined in the PIDR2 Register and the continuation code defined in the PIDR4 Register, gives the designer of the component.  This reflects either the targeted [4:1] from the DAP, or a sub-system identifier. |
| 3:0 | RO | 4’h8 | Bits [11:8] of the 12 bit part number of the component. The designer of the component assigns this part number.  This reflects either the targeted [27:24] from the DAP, or a sub-system identifier. |

Peripheral ID2 Register

Address: Operational Base + offset (0xFE8)

Table 306. Peripheral ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | An incremental value starting from 0b0000 for the first revision of this component. This increases by 1 for both major and minor revisions and is used to identify the major or minor revisions.  This reflects either the targeted [31:28] from the DAP, or a sub-system identifier. |
| 3 | RO | 1;b1 | Always set. Indicates if the JEDEC assigned designer ID is used.  0b1 Indicates that a JEDEC assigned designer ID is used. |
| 2:0 | RO | 4’b0 | Bits [6:4] of the JEDEC JEP106 identity code. This code, along with bits [3:0] of the identity code defined in the PIDR1 Register and the continuation code defined in the PIDR4 register, gives the designer of the component.  This reflects either the targeted [7:5] from the DAP, or a sub-system identifier. |

Peripheral ID3 Register

Address: Operational Base + offset (0xFEC)

Table 307. Peripheral ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Indicates minor errata fixes specific to the revision of the component being used, for example metal fixes after implementation. In most cases, this field is zero. ARM recommends that the component designers ensure that a metal fix can change this field if required, for example, by driving it from registers that reset to zero.  0b0000 Indicates that there are no errata fixes to this component. |
| 3:0 | RO | 4’b0 | Indicates if the customer has modified the behavior of the component. In most cases, this field is zero. The customer changes this value on modifications to this component.  0b0000 Indicates that the customer has not modified this component. |

Component ID0 Register

Address: Operational Base + offset (0xFF0)

Table 308. Component ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h0d | Contains bits [7:0] of the component identification code.  0x0D Bits [7:0] of the identification code. |

Component ID1 Register

Address: Operational Base + offset (0xFF4)

Table 309. Component ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Class of the component, for example, if the component is a ROM table or a generic CoreSight component. Contains bits [15:12] of the component identification code.  b0001 Indicates that the component is a ROM table. |
| 3:0 | RO | 4’b0 | Contains bits [11:8] of the component identification code.  0b0000 Bits [11:8] of the identification code. |

Component ID2 Register

Address: Operational Base + offset (0xFF8)

Table 310. Component ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b05 | Contains bits [23:16] of the component identification code.  0x05 Bits [23:16] of the identification code |

Component ID3 Register

Address: Operational Base + offset (0xFFC)

Table 311. Component ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’hb1 | Contains bits [31:24] of the component identification code.  0xB1 Bits [31:24] of the identification code. |

* + 1. TimeStamp Register Descriptions

Counter Control Register

Address: Operational Base + offset (0x000)

Table 312. Counter Control Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | RW | 1’b0 | Halt on Debug. The possible values are:  0 Halt on debug, HLTDBG signal into the counter has no effect.  1 Halt on debug, HLTDBG signal into the counter halts the counter. |
| 0 | RW | 1’b0 | Enable. The possible values are:  0 The counter is disabled and not incrementing.  1 The counter is enabled and is incrementing. |

Counter Status Register

Address: Operational Base + offset (0x004)

Table 313. Counter Status Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | Reserved | - | Reserved |
| 1 | RO | 4’b0 | Debug Halted |
| 0 | Reserved | - | Reserved. |

Current Counter Value Lower register

Address: Operational Base + offset (0x008)

Table 314. Current Counter Value Lower register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’b0 | Current value of the timestamp counter, lower 32bits. To change the current timestamp value, write the lower 32 bits of the new value to this register before writing the upper 32 bits to CNTCVU. The timestamp value is not changed until the CNTCVU register is written to. |

CNTFID0 Register

Address: Operational Base + offset (0x020)

Table 315. CNTFID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’b0 | Current value of the timestamp counter, lower 32bits. To change the current timestamp value, write the lower 32 bits of the new value to this register before writing the upper 32 bits to CNTCVU. The timestamp value is not changed until the CNTCVU register is written to. |

CNTCVU Register

Address: Operational Base + offset (0x00C)

Table 316. CNTCVU Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 32’b0 | Current value of the timestamp counter, upper 32 bits. To change the current timestamp value, write the lower 32 bits of the new value to CNTCVL before writing the upper 32 bits to this register. The 64-bit timestamp value is updated with the value from both writes when this register is written to. |

Peripheral ID4 Register

Address: Operational Base + offset (0xFD0)

Table 317. Peripheral ID4 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’h1 | This is a 4-bit value that indicates the total contiguous size of the memory window used by this component in powers of 2 from the standard 4KB. For example:  b0000 Indicates that the device occupies 4KB of memory.  b0001 Indicates that the device occupies 8KB of memory.  b0010 Indicates that the device occupies 16KB of memory.  b0011 Indicates that the device occupies 32KB of memory.  b0100 Indicates that the device occupies 64KB of memory.  b0101 Indicates that the device occupies 128KB of memory.  b0110 Indicates that the device occupies 256KB of memory.  b0111 Indicates that the device occupies 512KB of memory.  b1000 Indicates that the device occupies 1MB of memory.  b1001 Indicates that the device occupies 2MB of memory.  b1010 Indicates that the device occupies 4MB of memory.  b1011 Indicates that the device occupies 8MB of memory.  b1100 Indicates that the device occupies 16MB of memory.  b1101 Indicates that the device occupies 32MB of memory.  b1110 Indicates that the device occupies 64MB of memory.  The possible value is:  b0000 Indicates that the device only occupies 4KB of memory. |
| 3:0 | RO | 4’h2 | This is the JEDEC JEP106 continuation code. This code, along with bits[6:4] of the identity code defined in the PIDR0 Register and bits[3:0] of the identity code defined in the PIDR1 Register, gives the designer of the component.  0b0100 JEDEC continuation code. |

Peripheral ID5-7 Registers

Address: Operational Base + offset (0xFD4 0xFD8 0xFDC)

Table 318. Peripheral ID5-7 Registers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |

Peripheral ID0 Register

Address: Operational Base + offset (0xFE0)

Table 319. Peripheral ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h8c | Bits[7:0] of the 12 bit part number of the component. The designer of the component assigns this part number. This reflects either targeted [23:16] from the DAP, or a sub-system identifier. |

Peripheral ID1 Register

Address: Operational Base + offset (0XFE4)

Table 320. Peripheral ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | - | Reserved |
| 7:4 | RO | 4’he | Bits[3:0] of the JEDEC JEP106 identity code. This code, along with bits[6:4] of the identity code defined in the PIDR2 Register and the continuation code defined in the PIDR4 Register, gives the designer of the component.  This reflects either the targeted [4:1] from the DAP, or a sub-system identifier. |
| 3:0 | RO | 4’h8 | Bits[11:8] of the 12 bit part number of the component. The designer of the component assigns this part number.  This reflects either the targeted [27:24] from the DAP, or a sub-system identifier. |

Peripheral ID2 Register

Address: Operational Base + offset (0xFE8)

Table 321. Peripheral ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | An incremental value starting from 0b0000 for the first revision of this component. This increases by 1 for both major and minor revisions and is used to identify the major or minor revisions.  This reflects either the targeted [31:28] from the DAP, or a sub-system identifier. |
| 3 | RO | 1;b1 | Always set. Indicates if the JEDEC assigned designer ID is used.  0b1 Indicates that a JEDEC assigned designer ID is used. |
| 2:0 | RO | 4’b0 | Bits [6:4] of the JEDEC JEP106 identity code. This code, along with bits [3:0] of the identity code defined in the PIDR1 Register and the continuation code defined in the PIDR4 register, gives the designer of the component.  This reflects either the targeted [7:5] from the DAP, or a sub-system identifier. |

Peripheral ID3 Register

Address: Operational Base + offset (0xFEC)

Table 322. Peripheral ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Indicates minor errata fixes specific to the revision of the component being used, for example metal fixes after implementation. In most cases, this field is zero. ARM recommends that the component designers ensure that a metal fix can change this field if required, for example, by driving it from registers that reset to zero.  0b0000 Indicates that there are no errata fixes to this component. |
| 3:0 | RO | 4’b0 | Indicates if the customer has modified the behavior of the component. In most cases, this field is zero. The customer changes this value on modifications to this component.  0b0000 Indicates that the customer has not modified this component. |

Component ID0 Register

Address: Operational Base + offset (0xFF0)

Table 323. Component ID0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’h0d | Contains bits [7:0] of the component identification code.  0x0D Bits [7:0] of the identification code. |

Component ID1 Register

Address: Operational Base + offset (0xFF4)

Table 324. Component ID1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:4 | RO | 4’b0 | Class of the component, for example, if the component is a ROM table or a generic CoreSight component. Contains bits [15:12] of the component identification code.  b0001 Indicates that the component is a ROM table. |
| 3:0 | RO | 4’b0 | Contains bits [11:8] of the component identification code.  0b0000 Bits [11:8] of the identification code. |

Component ID2 Register

Address: Operational Base + offset (0xFF8)

Table 325. Component ID2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’b05 | Contains bits[23:16] of the component identification code.  0x05 Bits[23:16] of the identification code |

Component ID3 Register

Address: Operational Base + offset (0xFFC)

Table 326. Component ID3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | Reserved | - | Reserved |
| 7:0 | RO | 8’hb1 | Contains bits [31:24] of the component identification code.  0xB1 Bits [31:24] of the identification code. |

* 1. Programming Sequence



Figure 3 Coresight Debug mode



Figure 4. Coresight Trace mode

1. SPI-NOR Controller(TBU)
   1. Register Summary

Table .SPINOR Controller Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| SPI\_TxR | 0x0000 | 0x00000000 | SPI master/slave controller transmit FIFO input |
| SPI\_RxR | 0x0000 | 0x00000000 | SPI master/slave controller receiver FIFO output |
| SPI\_IER | 0x0004 | 0x00000000 | Enable/Mask interrupts generated by the SPI master controller |
| SPI\_FCR | 0x0008 | 0x00000000 | SPI master/slave controller FIFO control register |
| SPI\_FWCR | 0x000C | 0x00000100 | SPI master/slave controller transaction flow control register |
| SPI\_DLYCR | 0x0010 | 0x00000000 | SPI master controller delay control register (master only) |
| SPI\_TxCR | 0x0014 | 0x00000000 | Transmit counter (master only) |
| SPI\_RxCR | 0x0018 | 0x00000000 | Receive counter (master only) |
| SPI\_SSCR | 0x001C | 0x00000000 | SPI master/slave controller slave select and characteristic register |
| SPI\_ISR | 0x0020 | 0x00000000 | SPI master/slave controller interrupt status register |
| SPI\_FIFO\_STAT | 0x0024 | 0x00000011 | SPI Transmit FIFO and Receive FIFO status |
| SPI\_TX\_REG0 | 0x0028 | 0x00000000 | SPI transmit register 0 for normal usage |
| SPI\_TX\_REG1 | 0x002C | 0x00000000 | SPI transmit register 1 for normal usage |
| SPI\_TX\_REG2 | 0x0030 | 0x00000000 | SPI transmit register 2 for normal usage |
| SPI\_TX\_REG3 | 0x0034 | 0x00000000 | SPI transmit register 3 for normal usage |
| SPI\_RX\_REG0 | 0x0038 | 0x00000000 | SPI receive register 0 for normal usage |
| SPI\_RX\_REG1 | 0x003C | 0x00000000 | SPI receive register 1 for normal usage |
| SPI\_RX\_REG2 | 0x0040 | 0x00000000 | SPI receive register 2 for normal usage |
| SPI\_RX\_REG3 | 0x0044 | 0x00000000 | SPI receive register 3 for normal usage |
| SPI\_DMA\_REG | 0x0048 | 0x00000000 | SPI DMA configure regsiter |
| SPI\_BOOT\_CR | 0x004C | 0x00000000 | Bootable SPI control register |

* 1. Register Description

SPI\_TxR: This register contains data to be transmitted on the SPI master/slave controller bus on the MOSI pin

Address: Operational Base + offset (0x0000)

Table 328.SPI\_TxR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | WO | 0x0 | SPI master/slave controller transmit data FIFO input. |

SPI\_RxR:This register contains data received from the SPI master/slave controller bus on the MISO pin

Address: Operational Base + offset (0x00)

Table 329.SPI\_RxR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | RO | 0x0 | SPI master/slave controller receive data FIFO output. |

SPI\_IER: This register contains bits for controlling the interrupt generation of the SPI master/slave controller

Address: Operational Base + offset (0x04)

Table 330.SPI\_IER

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | - | - | Reserved |
| 11 | RW | 0x0 | Character length mismatch interrupt enable bit. (CHARLNIEN) (slave only)  The bit enables character length mismatch interrupt. When finishing transfer, the received data that character length mismatch the setting of SPI\_SSCR[14:11].  “1” Enabled.  “0” Disabled. |
| 10 | RW | 0x0 | Rx register 3 data available interrupt enable bit. (RXREG3IEN) (slave only)  The bit enables the Rx register 3 data available interrupt when one piece of data is available in Rx register 3.  “1” Enabled.  “0” Disabled. |
| 9 | RW | 0x0 | Rx register 2 data available interrupt enable bit. (RXREG2IEN) (slave only)  The bit enables the Rx register 2 data available interrupt when one piece of data is available in Rx register 2.  “1” Enabled.  “0” Disabled. |
| 8 | RW | 0x0 | Rx register 1 data available interrupt enable bit. (RXREG1IEN) (slave only)  The bit enables the Rx register 1 data available interrupt when one piece of data is available in Rx register 1.  “1” Enabled.  “0” Disabled. |
| 7 | RW | 0x0 | Rx register 0 data available interrupt enable bit. (RXREG0IEN) (Slave only)  The bit enables the Rx register 0 data available interrupt when one piece of data is available in Rx register 0.  “1” Enabled.  “0” Disabled. |
| 6 | RW | 0x0 | Transmit FIFO empty interrupt enable bit. (TXFEIEN)  The bit enables transmit FIFO interrupt when transmit FIFO is empty.  “1” Enabled.  “0” Disabled. |
| 5 | RW | 0x0 | Slave select signal rising detection enable bit. (SSNRIEN) (Slave only)  The bit enables the slave select signal rising detection interrupt when the master set slave select bit changes from “0” to “1”.  “1” Enabled.  “0” Disabled. |
| 4 | RW | 0x0 | Receive data available enable bit. (RXAVIEN)  The bit enables receive FIFO interrupt when at least one data available in receive FIFO.  “1” Enabled.  “0” Disabled. |
| 3 | RW | 0x0 | Transmit FIFO interrupt enable bit (TxFIEN). This bit enables the transmit FIFO interrupt when the transmit FIFO trigger level is reached.  “1” Enabled.  “0” Disabled. |
| 2 | RW | 0x0 | Receive FIFO interrupt enable bit (RxFIEN). This bit enables the receive FIFO interrupt when the receive FIFO trigger level is reached.  “1” Enabled.  “0” Disabled. |
| 1 | RW | 0x0 | Receive FIFO overrun interrupt enable bit (RxFOIEN). This bit enables the receive FIFO overrun interrupt when the receive FIFO overrun condition has occurred.  “1” Enabled.  “0” Disabled. |
| 0 | RW | 0x0 | Receive transfer complete interrupt enable bit (RxCIEN). (master only)  This bit enables the receive transfer complete interrupt each time a receive transaction has ended.  “1” Enabled.  “0” Disabled. |

SPI\_FCR: The FCR allows selection of the FIFO trigger level (the number of entries in the receive FIFO required to enable the receive FIFO interrupt and the number of empty entries in the transmit FIFO required to enable the transmit FIFO interrupt). The FIFOs can also be cleared using this register

Address: Operational Base + offset (0x08)

Table 331.SPI\_FCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:14 | - | - | Reserved |
| 13:11 | RW | 0x0 | Defines the receive FIFO interrupt trigger level.  The receive FIFO interrupt trigger level meaning is described below.  For example, trigger level 4 entries indicate that there are at 4 data available entries in the receive FIFO.  0x0 – 2 entries 0x1 – 4 entries  0x2 – 6 entries  0x3~0x7 – Reserved. |
| 10:8 | RW | 0x0 | Define the transmit FIFO interrupt trigger level.  The transmit FIFO interrupt trigger level meaning is described below.  For example, trigger level 4 entries indicate that there are at least 4 empty location entries for pushing data into the transmit FIFO.  0x0 – 2 entries 0x1 – 4 entries  0x2 – 6 entries  0x3~0x7 – Reserved. |
| 7:4 | - | - | Reserved. |
| 3 | WO | 0x0 | SPI master/slave controller receive FIFO reset bit (CLRRXF\_N).  Writing a ‘1’ to this bit resets the receive FIFO. |
| 2 | WO | 0x0 | SPI master/slave controller transmit FIFO reset bit (CLRTXF\_N).  Writing a ‘1’ to this bit resets the transmit FIFO. |
| 1 | RO | 0x0 | Transmit FIFO full flag (TxFF).  This bit is set whenever the transmit FIFO is full. |
| 0 | RO | 0x0 | Receive data available flag (RxDAF).  This bit is set whenever at least has one data entry is available in the receive FIFO. |

Notes:The transmit and receive FIFO reset signals sustain 3 pclk cycles. During the reset period, any access to FIFO is ignored.

SPI\_FWCR: This register is used to control SPI master/slave controller transaction flow

Address: Operational Base + offset (0x0C)

Table 332.SPI\_FWCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:15 | - | - | Reserved |
| 14 | RW | 0x0 | Slave select signal control enable bit (SSC) (master only)  0: Slave select signal is controlled by HW  1: Slave select signal is controlled by SW |
| 13 | RW | 0x1 | Slave select signal active bit (SSA) (master R/W, and slave read only)  Writing a ‘1’ to this bit will set the slave select signal not active. The value of slave select signal changes from ‘0’ to ‘1’.  Writing a ‘0’ to this bit will set the slave select signal active. The value of the slave select signal changes from ‘1’ to ‘0’. |
| 12 | RW | 0x0 | SPI master/slave mode (SPIMD). This bit selects the SPI master/slave mode.  “0” SPI is in Slave mode  “1” SPI is in Master mode |
| 11 | WO | 0x0 | SPI master/slave controller soft reset bit (SRST\_N). Writing a ‘1’ to this bit resets the SPI master/slave controller logic. |
| 10 | RW | 0x0 | SPI master/slave enable bit (SPIEN). This bit enables the SPI master controller.  “1” SPI master/slave enabled  “0” SPI master/slave disabled |
| 9 | RW | 0x0 | SPI run bit (SPIRUN). (Master only)  When the CPU sets this bit from “0” to “1”, the SPI master/slave controller begins to transfer the data stored in the transmit FIFO and/or receive the data into the receive FIFO on the SPI master/slave controller bus. This will automatically be cleared to “0” after the transaction has ended. |
| 8 | RW | 0x1 | SPI master controller clock idle enable bit (CKIDLEN). (Master only)  This bit determines whether or not the SPI master controller clock can be asserted in an idle state during a transaction process, assuming that the master core meets the receive FIFO full or transmit FIFO empty conditions.  “1” SPI master controller clock can be asserted in an idle state.  “0” SPI master controller clock cannot be asserted in an idle state. |
| 7 | - | - | Reserved. |
| 6 | RW | 0x0 | Hardware DMA request enable |
| 5 | RW | 0x0 | Transmit and receive simultaneously transfer enable (TxRxsten) (master only)  This bit indicates whether or not the transmit and receive transfer can occur concurrently during a transaction.  “0” Tx and Rx cannot concurrently happen.  ”1” Tx and Rx can concurrently happen. |
| 4 | RW | 0x0 | SPI master/slave controller clock polarity bit (CPOL). This bit determines the polarity of the SCLK.  “0” SCLK is low when idle.  “1” SCLK is high when idle. |
| 3 | RW | 0x0 | Clock Phase Bit (CPHA). This bit determines the clock phase of the SCLK in relationship to the serial data.  "0" data is valid on the first SCLK edge (rising or falling) after slave select has asserted.  "1" data is valid on the second SCLK edge (rising or falling) after slave select has asserted. |
| 2 | RW | 0x0 | LSB-First Enable (LSBEN).  “1” LSB first transfer.  “0” MSB first transfer. |
| 1 | - | - | Reserved. |
| 0 | RW | 0x0 | Loop back mode (LBKMD) (master only)  This bit is indicates the operation mode of the SPI master controller is in a normal operation mode or in a loop back mode.  ‘0’ Normal operation mode.  ‘1’ Loop back mode. |

Notes: The soft reset signal can sustain 3 pclk cycles. During the reset period, any access to FIFO is ignored.

SPI\_DLYCR :This register sets the necessary clock delay during a transaction according to specific slave device requirements

Address: Operational Base + offset (0x10)

Table 333.SPI\_DLYCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:11 | - | - | Reserved |
| 10:8 | RW | 0x0 | Period between Tx and Rx transfer (PBTxRx). This field defines the delay between transmit transfer complete and receive transfer start.  0x0: Non-SPI master controller clock delay.  0x1: 4 SPI master controller clock delay.  0x2: 8 SPI master controller clock delay.  0x3: 16 SPI master controller clock delay.  0x4: 32 SPI master controller clock delay.  0x5: 64SPI master controller clock delay.  0x6: 128 SPI master controller clock delay.  0x7: 256 SPI master controller clock delay. |
| 7:6 | - | - | Reserved. |
| 5:3 | RW | 0x0 | Period between two consecutive transfers (PBCT). This field defines the delay between consecutive transfers to the device without removing its chip select.  0x0: Non-SPI master controller clock delay.  0x1: 4 SPI master controller clock delay.  0x2: 8 SPI master controller clock delay.  0x3: 16 SPI master controller clock delay.  0x4: 32 SPI master controller clock delay.  0x5: 64 SPI master controller clock delay.  0x6: 128 SPI master controller clock delay.  0x7: 256 SPI master controller clock delay. |
| 2:0 | RW | 0x0 | Period before SPI master controller clock active (PBCA). This field defines the delay before the SPI master controller clock changes from idle to active after the chip select is asserted.  0x0: 1/2 SPI master controller clock delay.  0x1: 4 SPI master controller clock delay.  0x2: 8 SPI master controller clock delay.  0x3: 16 SPI master controller clock delay.  0x4: 32 SPI master controller clock delay.  0x5: 64SPI master controller clock delay.  0x6: 128 SPI master controller clock delay.  0x7: 256 SPI master controller clock delay. |

Notes: The timing diagram below illustrates the delay meaning.



Figure 5.SPI clock’s delay meaning

SPI\_TxCR: This register controls the total transfer data size in each transmit transaction

Address: Operational Base + offset (0x0014)

Table 334.SPI\_TxCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0 | 0: Stop the transmit transaction.  1-65535: Start a transmit transaction. |

SPI\_RxCR: This register controls the total transfer data size in each receive transaction

Address: Operational Base + offset (0x0018)

Table 335.SPI\_RxCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved. |
| 15:0 | RW | 0x0 | 0: Stop the receive transaction.  1-65535: Start a receive transaction. |

SPI\_SSCR: SPI master/slave controller slaves select and characteristic control register

Address: Operational Base + offset (0x001C)

Table 336.SPI\_SSCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:15 | - | - | Reserved. |
| 14:11 | RW | 0x0 | Character length determines bits (SPICHRL).  This field specifies how many bits are to be transferred in each transfer.  0x0: Reserved  0x1: Reserved  0x2: Reserved  0x3: 4 bits (master only)  0x4: 5 bits (master only)  0x5: 6 bits (master only)  0x6: 7 bits (master only)  0x7: 8 bits  0x8: 9 bits  0x9: 10 bits  0xA: 11 bits  0xB: 12 bits  0xC: 13 bits  0xD: 14 bits  0xE: 15 bits  0xF: 16 bits |
| 10:8 | RW | 0x0 | SPI master controller slave select register (SPISSR) (master only)  0x0: Slave0 device (SPIx\_SSN[0]) is set to the active state.  0x1: Slave1 device (SPIx\_SSN[1]) is set to the active state.  0x2: Slave2 device (SPIx\_SSN[2]) is set to the active state.  0x3: Slave3 device (SPIx\_SSN[3]) is set to the active state.  0x4: Slave4 device (SPIx\_SSN[4]) is set to the active state.  0x5: Slave5 device (SPIx\_SSN[5]) is set to the active state.  0x6: Slave6 device (SPIx\_SSN[6]) is set to the active state.  0x7: Slave7 device (SPIx\_SSN[7]) is set to the active state. |
| 7:6 | - | - | Reserved. |
| 5:0 | RW | 0x0 | SPI master controller clock divisor bits (SPIDIVR) (master only)  The value of SPIDIVR is used to generate the transmit and receive bit rate of the SPI master controller. The following section describes the bit rate equation in more detail. |

Notes: CPHA, CPOL, should be set to match the protocol expected by the SPI slave device.

SPI\_ISR: SPI master/slave controller interrupt status register

Address: Operational Base + offset (0x0020)

Table 337.SPI\_ISR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | - | - | Reserved. |
| 11 | RW | 0x0 | Character length mismatch interrupt flag (CHARLNIF) (slave only).  While finishing transfer, the received data that character length mismatch the setting of SPI\_SSCR[14:11], the bit is set.  An interrupt will be asserted to the CPU when this bit is set and the CHARLNIEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 10 | RW | 0x0 | Rx register 3 data available interrupt flag (RXREG3IF) (slave only).  This bit is set when data available in Rx register 3. An interrupt will be asserted to the CPU when this bit is set and the RXREG3IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 9 | RW | 0x0 | Rx register 2 data available interrupt flag (RXREG2IF) (slave only).  This bit is set when data is available in Rx register 2. An interrupt will be asserted to the CPU when this bit is set and the RXREG2IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 8 | RW | 0x0 | Rx register 1 data available interrupt flag (RXREG1IF) (slave only).  This bit is set when data is available in Rx register 1. An interrupt will be asserted to the CPU when this bit is set and the RXREG1IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 7 | RW | 0x0 | Rx register 0 data available interrupt flag (RXREG0IF) (slave only).  This bit is set when data is available in Rx register 0. An interrupt will be asserted to the CPU when this bit is set and the RXREG0IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 6 | RO | 0x0 | Transmit FIFO empty interrupt flag (TXFEIF).  This bit is set when the transmit FIFO is empty. An interrupt will be asserted to the CPU when this bit is set and the TXFEIEN bit is enabled. This bit will be cleared when the transmit FIFO is not empty. |
| 5 | RW | 0x0 | Slave select signal rising interrupt flag (SSNRIF) (slave only).  This bit is set when the slave select signal is asserted. An interrupt will be asserted to the CPU when this bit is set and the SSNRIEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 4 | RO | 0x0 | Receive data available interrupt flag (RXAVIF).  This bit is set when at least one piece of data is available in receive FIFO. An interrupt will be asserted to the CPU when this bit is set and the RXAVIEN bit is enabled. This bit will be cleared when the receive FIFO is empty. |
| 3 | RO | 0x0 | Transmit FIFO interrupt flag (TxFIF).  This bit is set when the transmit FIFO trigger level is reached, and CPU wishes to keep transmit data to the device. An interrupt will be asserted to the CPU when this bit is set and the TxFIEN bit is enabled. This bit will be cleared when the transmit FIFO pointer over the trigger level. |
| 2 | RO | 0x0 | Receive FIFO interrupt flag (RxFIF).  This bit is set whenever the receive FIFO trigger level is reached. An interrupt will be asserted to the CPU when this bit is set and the RxFIEN bit is enabled. This bit will be cleared when the receive FIFO pointer drops below the trigger level. |
| 1 | RO | 0x0 | SPI master/slave controller overrun interrupt flag (SPIORIF).  ‘1’ – If the receive FIFO is full and another character has been received in the receiver shift register. If another character is starting to arrive, it will overwrite the data in the shift register, but the FIFO will remain intact. The bit is cleared if the receive FIFO is cleared by software simultaneously.  ‘0’ – No overrun state an another SPI master controller transaction |
| 0 | RO | 0x0 | Receive complete interrupt flag (RxCIF) (master only).  This bit is set whenever the receive transaction is over. An interrupt will be asserted to the CPU when this bit is set and the RxCIEN bit is enabled. The bit is cleared upon reading from the register. |

SPI\_FIFO\_STAT: This register contains the SPI Transmit FIFO and Receive FIFO status

Address: Operational Base + offset (0x0024)

Table 338.SPI\_ FIFO\_STAT

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:7 | - | - | Reserved. |
| 6 | RO | 0x0 | Receive FIFO full bit. (RxFF\_FULL)  This bit shows whether or not the Receive FIFO is full.  “1”: Receive FIFO is full.  “0”: Receive FIFO is not full. |
| 5 | RO | 0x0 | Receive FIFO half full bit. (RxFF\_HALFFULL)  This bit shows whether or not the Receive FIFO is half full.  “1”: There are four pieces of data in the Receive FIFO.  “0”: There are not four pieces of data in the Receive FIFO. |
| 4 | RO | 0x1 | Receive FIFO empty bit. (RxFF\_EMPTY)  This bit shows whether or not the Receive FIFO is empty.  “1”: Receive FIFO is empty.  “0”: Receive FIFO is not empty. |
| 3 | - | - | Reserved. |
| 2 | RO | 0x0 | Transmit FIFO full bit. (TxFF\_FULL)  This bit shows whether or not the transmit FIFO is full.  “1”: Transmit FIFO is full.  “0”: Transmit FIFO is not full. |
| 1 | RO | 0x0 | Transmit FIFO half full bit. (TxFF\_HALFFULL)  This bit shows whether or not the transmit FIFO is half full.  “1”: There are four or more pieces of data in the Transmit FIFO.  “0”: There are not four pieces of data in the Transmit FIFO. |
| 0 | RO | 0x1 | Transmit FIFO empty bit. (TxFF\_EMPTY)  This bit shows whether or not the transmit FIFO is empty.  “1”: Transmit FIFO is empty.  “0”: Transmit FIFO is not empty. |

Note: The half full bit will also be “1” when FIFO if full.

SPI\_TX\_REGx (x = 0, 1, 2, 3): This register contains data to be transmitted on the SPI controller bus on the MISO pin

Address: Operational Base + offset (0x0028/0x002C/0x0030/0x0034)

Table 339.SPI\_ TX\_REGx (x = 0, 1, 2, 3)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved. |
| 15:0 | RW | 0x0 | SPI master/slave controller transmit data register (slave only) |

SPI\_RX\_REGx (x = 0, 1, 2, 3): This register contains the data received from the SPI controller bus on the MOSI pin

Address: Operational Base + offset (0x0038/0x003C/0x0040/0x0044)

Table 340.SPI\_ RX\_REGx (x = 0, 1, 2, 3)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved. |
| 15:0 | RO | 0x0 | SPI master/slave controller receive data register (slave only) |

SPI\_DMA\_REG**:** This register contains the control bit to select DMA mode.

Address: Operational Base + offset (0x0048)

Table 341. SPI\_BOOT\_CR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:3 | - | - | Reserved. |
| 2 | RW | 0x0 | DMA burst mode select  0x0: TX Single;  0x1: TX Burst; |
| 1 | RW | 0x0 | DMA burst mode select  0x0: RX Single;  0x1: RX Burst; |
| 0 | RW | 0x0 | DMA interface select  0x0: ARM DMA interface;  0x1: Socle DMA interface; |

SPI\_BOOT\_CR**:** This register contains the control bit to control SCLK (divided from HCLK) frequency and boot enable.

Address: Operational Base + offset (0x004C)

Table 342. SPI\_BOOT\_CR

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Attr** | **Reset Value** | **Description** |
| 31:7 | - | - | Reserved. |
| 6 | RW | 0x1 | Boot enable  0x0:APB interface can access SPI, for legacy SPI  0x1: AHB interface can access SPI, support bootable function  When we need to set register in legacy APB-interfaced SPI, this bit should be set to 0x0 |
| 5:0 | RW | 0x0 | SPI master controller clock divisor bits (SPIDIVR)  The value of SPIDIVR is used to generate the transmit and receive bit rate of the SPI master controller. The following section describes the bit rate equation in more detail.  The frequency divisor equation for generation sclk is below:  sclk Divisor = (SPIDIVR[5:3] + 1) x 2(SPIDIVR[2:0] + 1)  sclk = hclk/ sclk Divisor |

* 1. Programming Sequence
     1. SPI master/slave controller operation flow chart

The flow chart below describes how the software configure and performs a SPI master controller transaction through the SPI master, APB write data flow.

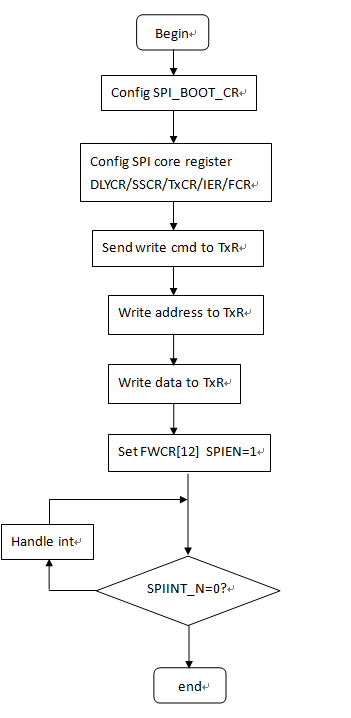


Figure 6. APB write flow

* + 1. Read Data

Apb read : Directly to read data by send command to spi device.

Ahb read: Config 1/2/4bit data & 3/4byte address transmit by the SCU. The address of ahb read data is IP\_ahb\_base + data address.

Note: If needs to read data by AHB bus , its should config SPI\_BOOT\_CR register firstly, open ahb read enble.

The flow chart below describes how the software configures and performs a APB read data .

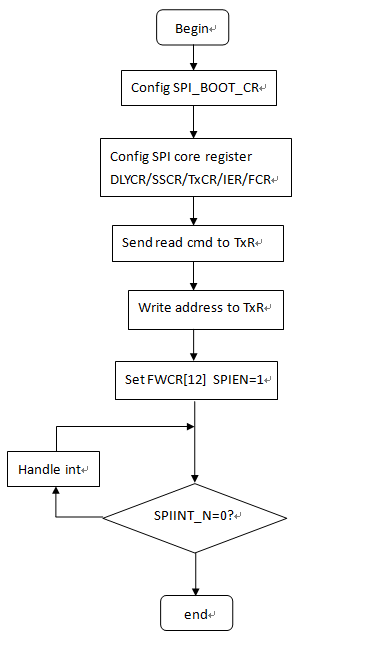


Figure 7. APB read data

The flow chart below describes how the software configures and performs a AHB read data .

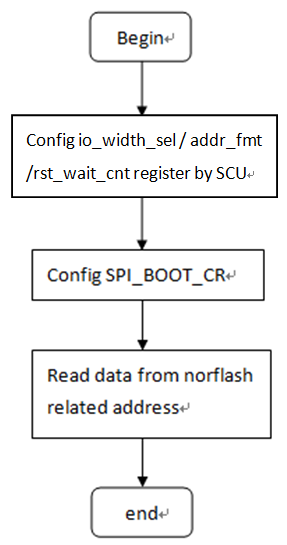


Figure 8. AHB read flow

1. IVA Acceleration
   1. Register Summary

Table 343.IVA AccelerationRegister summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| SI | 0x00000~0x007FF | UNK | IP\_CONFIG |
| AVIF | 0x00800~0x0087F | UNK | IF\_CONFIG |
| GAMMA\_FE0\_MEM | 0x01000~0x010FF | UNK | GAMMA\_FE0\_MEM |
| GAMMA\_FE1\_MEM | 0x01800~0x01FFF | UNK | GAMMA\_FE1\_MEM |
| METERING\_MEM | 0x02000~0x02FFF | UNK | METERING\_MEM |
| HISTOGRAM\_MEM | 0x03000~0x033FF | UNK | HISTOGRAM\_MEM |
| CMD\_QUEUES | 0x03800~0x03BFF | UNK | CMD\_QUEUES |
| AV\_CORE | 0x04000~0x04FFF | UNK | IP\_CONFIG\_ANALYTIX |

* 1. Register Descriptions
     1. SI Registers

ID

Address: Operational Base + offset 0x0\_0000

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | UNK | API[31:0] |

Top

Address: Operational Base + offset 0x00010

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| [31:0] | RW | 0x00000780 | | Active Width Active video width in pixels |

Address: Operational Base + offset 0x00014

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| [31:0] | RW | 0x00000438 | | Active Height Active video height in lines |

Address: Operational Base + offset 0x00018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| [31:3] | RW | 0x0 | | -- |
| [2:0] | RW | 0x0 | | RGGB start  0 = R Gr  Gb B    1 = Gr R  B Gb    2 = Gb B  R Gr    3 = B Gb  Gr R |

Address: Operational Base + offset 0x00028

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x 0 | -- |
| [15:0] | RW | 0x20 | Flush hblank  Horizontal blanking interval during regeneration (0=measured input interval) |

Address: Operational Base + offset 0x00030

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:2] | RW | 0x 0 | -- |
| [1:0] | RW | 0x 0 | Config Buffer Mode  0 = Disabled (config updates immediately)  1 = Blocked (config never updates)  2 = Local (module config updates during local vertical blanking)  3 = Global (all module config updated during ISP vertical blanking) |

Address: Operational Base + offset 0x00040

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:17] | RW | 0x 0 | -- |
| [16] | RW | 0x 0 | Bypass demosaic  Bypass demosaic module (output RAW data) |
| [15:6] | RW | 0x 0 | -- |
| [5] | RW | 0x 0 | Bypass RAW frontend Bypass RAW frontend (green equalization and dynamic defect pixel)  Bypass gamma fe Bypass WDR companded frontend lookup table |
| [4] | RW | 0x 0 | -- |
| [3] | RW | 0x 0 | Bypass sensor offset  Bypass frontend black level adjustment |
| [2] | RW | 0x 0 | -- |
| [1] | RW | 0x 0 | Bypass Video Test Gen |
| [0] | RW | 0x 0 | Bypass Video Test Pattern Generator Bypass Video Test Pattern Generator |

Address: Operational Base + offset 0x00044

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:15] | RW | 0x 0 | -- |
| [14:13] | RW | 0x 0 | Used to select data to DMA writer  0 = Select image sensor RAW data (from RAW bypass mux).  1 = Select image sensor RAW data (direct from input port)  2 = Processed RGB data  3 = Select image sensor RGB data. |
| [12] | RW | 0x 0 | Used to select between sensor raw input or RGB888 or YUV422 input  0 = Select RAW data (from RAW bypass mux).  1 = Select RGB data (from the image sensor and preprocessed to provide intensity to AE) |
| [11] | RW | 0x 0 | Used to bypass the RAW pipeline and provide RAW data to AE and the DMA writer  0 = Select data from RAW pipeline.  1 = Select image sensor RAW data. |
| [10] | RW | 0x 0 | Used to select video stream output to the analytics engine  0 = Select data from RAW pipeline.  1 = Select image sensor data (RAW or RGB). |
| [9] | RW | 0x 0 | Used to select data to Debayer function  0 = Select data from RAW pipeline.  1 = Select data from Memory. |
| [8] | RW | 0x 0 | Used to select data for RAW pipeline  0 = Select image sensor RAW data.  1 = Select data from Memory |
| [7:0] | RW | 0x 0 | -- |

Address: Operational Base + offset 0x00078

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [0] | RW | 0x0 | 1= synchronous reset of FSMs in design (faster recovery after broken frame) |

Interrupt controller

Address: Operational Base + offset 0x00080

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:21] | RW | 0x0 | -- |
| [20:16] | RW | 0x0 | Interrupt1 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |
| [15:5] | RW | 0x0 |  |
| [4:0] | RW | 0x0 | Interrupt0 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |

Address: Operational Base + offset 0x00084

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:21] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [20:16] | RW | 0x0 | Interrupt3 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |
| [15:5] | RW | 0x0 |  |
| [4:0] | RW | 0x0 | Interrupt2 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |

Address: Operational Base + offset 0x00088

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:21] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [20:16] | RW | 0x0 | Interrupt5source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |
| [15:5] | RW | 0x0 |  |
| [4:0] | RW | 0x0 | Interrupt4 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |

Address: Operational Base + offset 0x0008C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:21] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [20:16] | RW | 0x0 | Interrupt7 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |
| [15:5] | RW | 0x0 |  |
| [4:0] | RW | 0x0 | Interrupt6 source  Interrupt source selector. See ISP guide for interrupt definitions and valid values |

Address: Operational Base + offset 0x000A0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| [31:8] | RW | 0x0 | | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [7:0] | RW | 0x0 | | Interrupt status Interrupt event flags |

Address: Operational Base + offset 0x000A4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| [31:8] | RW | 0x0 | | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [7:0] | RW | 0x0 | | Interrupt clear  Interrupt event clear register writing 0-1 transition will clear the corresponding event flags. |

Input port Registers

Address: Operational Base + offset 0x00100

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| [31:25] | RW | 0x0 | | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [24] | RW | 0x0 | | 0 = don't invert acl\_i for acl gate  1 = invert acl\_i for acl gate |
| [23] | RW | 0x0 | | |  |  | | --- | --- | | |  | | --- | |  | |   0 = vertical counter is reset on rising edge of hs  1 = vertical counter is reset on rising edge of vs |
| [22:21] | RW | 0x0 | | -- |
| [20] | RW | 0x1 | | 0 = vertical counter increments on rising edge of HS  1 = vertical counter increments on falling edge of HS |
| [19] | RW | 0x0 | | 0 = horizontal counter is reset on rising edge of hs  1 = horizontal counter is reset on vsync (e.g. when hsync is not available) |
| [18] | RW | 0x0 | | 0 = don't invert polarity of HS for HS gate  1 = invert polarity of HS for HS gate |
| [17] | RW | 0x0 | | 0 = don't invert polarity of HS for ACL gate  1 = invert polarity of HS for ACL gate |
| [16] | RW | 0x0 | | 0 = normal mode  1 = hvalid = hsync XOR vsync0 = vertical counter counts on hs  1 = vertical counter counts on horizontal counter overflow or reset |
| [15] | RW | 0x0 | | 0 = vertical counter counts on hs  1 = vertical counter counts on horizontal counter overflow or reset |
| [14] | RW | 0x0 | | 0 = vertical counter counts on hs  1 = vertical counter counts on horizontal counter overflow or reset |
| [13] | RW | 0x0 | | -- |
| [12] | RW | 0x0 | | 0 = use hsync\_i port for active-line  1 = use acl\_i port for active-line |
| [11] | RW | 0x0 | | 0 = don't invert polarity for ACL gate  1 = invert polarity for ACL gate |
| [10] | RW | 0x0 | | 0 = horizontal counter reset on rising edge  1 = horizontal counter reset on falling edge |
| [9] | RW | 0x0 | | 0 = vsync is pulse-type  1 = vsync is toggle-type (field signal) |
| [8] | RW | 0x0 | | 0 = use vsync\_i port for vertical sync  1 = use field\_i port for vertical sync |
| [7:4] | RW | 0x0 | | -- |
| [3:0] | RW | 0x2 | | Allows selection of various input port presets for standard sensor inputs. See ISP Guide for details of available presets.  0-14: Frequently used presets. If using one of available presets, remaining bits in registers 0x100 and 0x104 are not used.  15: Input port configured according to registers in 0x100 and 0x104. Consult Apical support for special input port requirements |

Address: Operational Base + offset 0x00104

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [27] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | | 0 = pulse field  1 = toggle field | | |
| [26] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | |  | |   0 = exclude field\_i signal in Field gate  1 = include field\_i signal in Field gate |
| [25] | RW | 0x0 | 0 = exclude window2 signal in Field gate  1 = include window2 signal in Field gate |
| [24] | RW | 0x0 | 0 = exclude field\_i signal in Field gate  1 = include field\_i signal in Field gate |
| [23:20] | RW | 0x0 | -- |
| [19] | RW | 0x0 | 0 = exclude window2 signal in Field gate  1 = include window2 signal in Field gate |
| [18] | RW | 0x0 | 0 = exclude window1 signal in HS gate  1 = include window1 signal in HS gate |
| [17] | RW | 0x0 | 0 = exclude hsync signal in HS gate  1 = include hsync signal in HS gate |
| [16] | RW | 0x0 | 0 = exclude window1 signal in HS gate  1 = include window1 signal in HS gate |
| [15:13] | RW | 0x0 | -- |
| [12] | RW | 0x0 | 0 = exclude vsync signal in ACL gate  1 = include vsync signal in ACL gate |
| [11] | RW | 0x0 | 0 = exclude acl\_i signal in ACL gate  1 = include acl\_i signal in ACL gate |
| [10] | RW | 0x0 | 0 = exclude window2 signal in ACL gate  1 = include window2 signal in ACL gate |
| [9] | RW | 0x0 | 0 = exclude hsync signal in ACL gate  1 = include hsync signal in ACL gate |
| [8] | RW | 0x0 | 0 = exclude window0 signal in ACL gate  1 = include window0 signal in ACL gate |
| [7:2] | RW | 0x0 | -- |
| [1] | RW | 0x0 | 0 = field is pulse-type  1 = field is toggle-type |
| [0] | RW | 0x0 | 0 = don't invert field\_i for field gate  1 = invert field\_i for field gate |

Address: Operational Base + offset 0x00108

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | hc limit  horizontal counter limit value (counts:0,1,...hc\_limit-1,hc\_limit,0,1,...) |

Address: Operational Base + offset 0x0010C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | hc start0  window0 start for ACL gate. See ISP guide for further details | | | |

Address: Operational Base + offset 0x00110

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  | | --- | |  | |
| [15:0] | RW | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | hc size0  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x00114

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | hc start1  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x00118

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | hc size1  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x0011C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | vc limit  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x00120

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | vc start  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x00124

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | vc size  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x00128

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RO | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RO | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | frame width  window0 size for ACL gate. See ISP guide for further details. | | | | |

Address: Operational Base + offset 0x0012C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RO | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RO | 0x0 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | |  | frame height  Window0 size for ACL gate. See ISP guide for further details | | | | |

Address: Operational Base + offset 0x00130

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [7] | RW | 0x0 | |  |  |  | | --- | --- | --- | |  | | | |  |   Used to freeze input port configuration. Used when multiple register writes are required to change input port configuration.  0 = normal operation  1 = hold previous input port config state |
| [6:3] | RW | 0x0 | -- |
| [2:0] | RW |  | Used to stop and start input port. See ISP guide for further details.  0 = safe stop  1 = safe start  2 = urgent stop  3 = urgent start  4 = Reserved  5 = safer start  6 = Reserved  7 = Reserved |

Address: Operational Base + offset 0x00134

Sensor Offset

Address: Operational Base + offset 0x00140

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [11:0] | RW | 0x0 | |  | | --- | |  |   Black 00  Black offset for color channel 00 (R) |

Address: Operational Base + offset 0x00144

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [11:0] | RW | 0x0 | Black 01  Black offset for color channel 00 (R) |

Address: Operational Base + offset 0x00148

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [11:0] | RW | 0x0 | Black 10  Black offset for color channel 00 (R) |

Address: Operational Base + offset 0x0014C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [11:0] | RW | 0x0 | Black 1  Black offset for color channel 00 (R) |

Frontend LUT Registers

Address: Operational Base + offset 0x00160

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| [31:8] | RW | 0x0 | -- |
| [7] | RW | 0x0 | Offset Mode 1  Lookup1 reflection mode for black offset region  0 = Manual curve reflection  1 = Automatic curve reflection |
| [6] | RW | 0x0 | Offset Mode 0  Lookup0 reflection mode for black offset region  0 = Manual curve reflection  1 = Automatic curve reflection |
| [5] | RW | 0x0 | MCU ready1  LUT1 status indicator. When 1, LUT1 is ready to receive the data from CPU |
| [4] | RW | 0x0 | MCU ready0  LUT0 status indicator. When 1, LUT0 is ready to receive the data from CPU |
| [3] | RW | 0x0 | MCU priority  Priority of CPU writes to LUTs  0=low. CPU read/writes from/to the frontend LUTs are only executed when LUTs are not being accessed by ISP. Normal operation.  1=high. CPU read/writes from/to the frontend LUTs are always executed. This may result in corrupt image data and invalid read back and is not recommended |
| [2] | RW | 0x0 | -- |
| [1] | RW | 0x0 | Enable1 Frontend lookup1 enable: 0=off 1=on |
| [0] | RW | 0x0 | Enable0 Frontend lookup0 enable: 0=off 1=on |

RAW Frontend

Address: Operational Base + offset 0x00180

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| 7 | RW | 0x0 | Disable detection of bright pixels |
| 6 | RW | 0x0 | Disable detection of dark pixels |
| [5:4] | RW | 0x0 | -- |
| 3 | RW | 0x0 | Show Defect Pixel: 0=off, 1=on |
| 2 | RW | 0x0 | Dynamic Defect Pixel enable: 0=off, 1=on |
| 1 | RW | 0x0 | -- |
| 0 | RW | 0x0 | Green equalization enable: 0=off, 1=on |

Address: Operational Base + offset 0x00184

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [7:0] | RW | 0x0 | Controls strength of Green equalization. Set during calibration |

Address: Operational Base + offset 0x00188

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [15:0] | RW | 0x0 | |  | | --- | | Controls strength of Green equalization. Set during calibration | |

Address: Operational Base + offset 0x00184

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | --   |  |  | | --- | --- | | |  | | --- | |  | | |
| [7:0] | RW | 0x0 | Controls strength of Green equalization. Set during calibration |

Address: Operational Base + offset 0x00188

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | -- |
| [15:0] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | | Debug selection port | | |

Address: Operational Base + offset 0x0018C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | | Defect pixel threshold | | |

Address: Operational Base + offset 0x00190

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | |  | | | |  | | --- | | Green equalization threshold | | | | |

Address: Operational Base + offset 0x00194

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | |  | |   Slope for HP Mask function |

Address: Operational Base + offset 0x00198

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Controls the sensitivity of green equalization to edges |

Address: Operational Base + offset 0x0019C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x0 | |  | | --- | | Slope for GE Mask function | |

Address: Operational Base + offset 0x001A0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Controls the aggressiveness of the dynamic defect pixel correction near edges |

Address: Operational Base + offset 0x001A4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Controls the directional nature of the dynamic defect pixel correction near edges |

Address: Operational Base + offset 0x001A8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Controls the directional nature of the dynamic defect pixel correction near edges |

Address: Operational Base + offset 0x001AC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Manual override of noise estimation |

Address: Operational Base + offset 0x001B0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x0 | Noise threshold for long exposure data  Format: unsigned 4.4-bit floating-point |

Address: Operational Base + offset 0x001B4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x0 | Noise threshold for long exposure data  Format: unsigned 4.4-bit floating-point |

Noise Profile RAW frontend

Address: Operational Base + offset 0x001C0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | -- |
| [15:0] | RW | 0x0 | Noise threshold for long exposure data  Format: unsigned 4.4-bit floating-point |

Address: Operational Base + offset0x001C4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x0 | Multiplier applied to short exposure data for noise profile calculation  Format: unsigned 6.2-bit fixed-point |

Address: Operational Base + offset 0x001C8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x0 | |  |  | | --- | --- | | |  | | --- | | Multiplier applied to long exposure data for noise profile calculation  Format: unsigned 6.2-bit fixed-point | | |

Address: Operational Base + offset 0x001CC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| 7 | RW | 0x0 | Defines how values below black level are obtained.  0: Repeat the first table entry.  1: Reflect the noise profile curve below black level |
| [6:0] | RW | 0x0 | Noise profile black level offset |

Noise Profile RAW frontend LUT

Address: Operational Base + offset 0x00380

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | -- | Weight lut(3) |
| [23:16] | RW | -- | Weight lut(2) |
| [15:8] | RW | -- | Weight lut(1) |
| [7:0] | RW | -- | Weight lut(0) |

Address: Operational Base + offset 0x00384

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | -- | Weight lut(7) |
| [23:16] | RW | -- | Weight lut(6) |
| [15:8] | RW | -- | Weight lut(5) |
| [7:0] | RW | -- | Weight lut(4) |

... repeated until ...

Address: Operational Base + offset 0x003FC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | -- | Weight lut(127) |
| [23:16] | RW | -- | Weight lut(126) |
| [15:8] | RW | -- | Weight lut(125) |
| [7:0] | RW | -- | Weight lut(124) |

Note: This is an array of 128 8-bit registers. The address for element (i) is: 0x00380 + 4 \* int(i / 4)

Statistics

Address: Operational Base + offset 0x00200

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x 10 | Hist Thresh 0 1  Histogram threshold for bin 0/1 boundary |

Address: Operational Base + offset 0x00204

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x 20 | Hist Thresh 1 2  Histogram threshold for bin 1/2 boundary |

Address: Operational Base + offset 0x00208

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x00 | -- |
| [7:0] | RW | 0x D0 | Hist Thresh 3 4  Histogram threshold for bin 2/3 boundary |

Address: Operational Base + offset 0x0020C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x 00 | -- |
| [7:0] | RW | 0x E0 | Hist Thresh 4 5  Histogram threshold for bin 3/4 boundary |

Address: 0x00220

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | -- | -- |
| [15:0] | RW | Hist 0  Normalized histogram results for bin 0 |

Address: Operational Base + offset 0x00224

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | -- | -- |
| [15:0] | RW | Hist 1  Normalized histogram results for bin 1 |

Address: Operational Base + offset 0x00228

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | -- | -- |
| [15:0] | RW | Hist 3  Normalized histogram results for bin 3 |

Address: Operational Base + offset 0x0022C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | -- | -- |
| [15:0] | RW | Hist 4  Normalized histogram results for bin 4 |

Address: Operational Base + offset 0x00230

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x00 | -- |
| [15:8] | RW | 0x 0F | AEXP Nodes Used Vert  Number of active zones vertically for AE stats collection |
| [7:0] | RW | 0x 0F | AEXP Nodes Used Horiz  Number of active zones horizontally for AE stats collection |

Address: Operational Base + offset 0x00268

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x00 | -- |
| 0 | RW | 0x00 | AEXP Nodes Used Vert  Number of active zones vertically for AE stats collection |
| [7:0] | RW | 0x00 | AEXP Nodes Used Horiz  Number of active zones horizontally for AE stats collection |

Address: Operational Base + offset 0x00240

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:10] | RW | 0x0 | -- |
| [9:0] | RW | 0x0 | Black Level AWB  Upper limit of valid data for AWB |

Address: Operational Base + offset 0x00244

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:10] | RW | 0x0 | -- |
| [9:0] | RW | 0x0 | Black Level AWB  Lower limit of valid data for AWB |

Address: Operational Base + offset 0x00248

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x 1FF | Cr Ref Max AWB  Maximum value of R/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x0024C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x000 | -- |
| [11:0] | RW | 0x 040 | Cr Ref Max AWB  Minimum value of B/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x00250

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x000 | -- |
| [11:0] | RW | 0x 1FF | Cb Ref Min AWB  Maximum value of R/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x00254

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x 040 | Cb Ref Min AWB  Minimum value of R/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x00258

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | -- |
| [11:0] | RW | -- | AWB RG  AWB statistics R/G color ratio output  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x0025C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | -- |
| [11:0] | RW | AWB BG  AWB statistics B/G color ratio output  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x00260

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | AWB SUM  AWB output population. Number of pixels used for AWB statistics  Format: unsigned 32-bit integer |
| -- |

Address: Operational Base + offset 0x00270

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x00 | -- |
| [15:8] | RW | 0x 0F | AWB Nodes Used Vert  Number of active zones vertically for AWB stats |
| [7:0] |  | 0x 0F | AWB Nodes Used Horiz  Number of active zones horizontally for AWB stats |

Address: Operational Base + offset 0x002D0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x FFF | Cr Ref High AWB  Maximum value of R/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x002D4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Cr Ref Low AWB  Minimum value of R/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x002D8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x000000 | -- |
| [11:0] | RW | 0xFFF | Cb Ref High AWB  Maximum value of B/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x002DC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | Cb Ref Low AWB  Minimum value of B/G for white region  Format: unsigned 4.8-bit fixed-point |

Address: Operational Base + offset 0x002C0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| 7 | RW | 0x0 | Cb Ref Low AWB  Minimum value of B/G for white region  Format: unsigned 4.8-bit fixed-point |
| [6:4] | RW | 0x0 | skip y  Histogram decimation in vertical direction: 0=every pixel; 1=every 2nd pixel; 2=every 3rd pixel; 3=every 4th pixel; 4=every 5th pixel; 5=every 8th pixel ; 6+=every 9th pixel |
| 3 | RW | 0x0 | offset x  0= start from the first column; 1=start from second column |
| [2:0] | RW | 0x0 | skip x  Histogram decimation in horizontal direction: 0=every 2nd pixel; 1=every 3rd pixel; 2=every 4th pixel; 3=every 5th pixel; 4=every 8th pixel ; 5+=every 9th pixel |

Address: Operational Base + offset 0x002C4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | -- | -- |
| [7:4] | RW | -- | scale top [3:0]  scale of top half of the range: 0=1x ,1=2x, 2=4x, 4=8x, 4=16x |
| [3:0] | RW | -- | scale bottom [3:0]  scale of bottom half of the range: 0=1x ,1=2x, 2=4x, 4=8x, 4=16x |

Address: Operational Base + offset 0x002C8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | Total Pixels  Total number of pixels processed (skip x and skip y are taken into account) |

Address: Operational Base + offset 0x002CC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | Counted Pixels [31:0]  Number of pixels accumulated (with nonzero weight) |

DMA Writer

Address: Operational Base + offset 0x002CC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | Counted Pixels [31:0]  Number of pixels accumulated (with nonzero weight) |
| 11 | RW | 0x0 | axi xact comp  0 = dont wait for axi transaction completion at end of frame(just all transfers accepted). 1 = wait for all transactions completed |
| 10 | RW | 0x0 | half irate  0 = normal operation , 1= write half(alternate) of input frames( only valid for continuous mode) |
| 9 | RW | 0x0 | frame write on  0 = no frames written(when switched from 1, current frame completes writing before stopping),  1= write frame(s) (write single or continous frame(s) ) |
| 8 | RW | 0x0 | single frame  0 = All frames are written(after frame\_write\_on= 1), 1= only 1st frame written ( after frame\_write\_on =1) |
| [7:6] | RW | 0x0 | Plane select [1:0]  Plane select for planar base modes. Only used if planar outputs required. Not used. Should be set to 0 |
| 5 | RW | 0x0 |  |
| 4 | RW | 0x0 |  |
| [3:0] | RW | 0x0 | Base mode [3:0]  Base DMA packing mode for RGB/RAW/YUV etc (see ISP guide) |

Address: Operational Base + offset 0x00304

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0438 | active height [15:0]  Active video height in lines 128-8000 |
| [15:0] | RW | 0x 0780 | active width [15:0]  Active video width in pixels 128-8000 |

Address: Operational Base + offset 0x00308

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000000000 | bank0\_base [31:0]  bank 0 base address for frame buffer, should be word-aligned |

Address: Operational Base + offset 0x0030C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000000000 | Bank1\_base [31:0]  bank 1base address for frame buffer, should be word-aligned |

Address: Operational Base + offset 0x00310

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000000000 | Bank2\_base [31:0]  bank 2 base address for frame buffer, should be word-aligned |

Address: Operational Base + offset 0x00314

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000000000 | Bank3\_base [31:0]  bank 3base address for frame buffer, should be word-aligned |

Address: Operational Base + offset 0x00318

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000000000 | Bank4\_base [31:0]  bank 4base address for frame buffer, should be word-aligned |

Address: Operational Base + offset 0x0031C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RW | 0x | Bank4\_base [31:0]  bank 4base address for frame buffer, should be word-aligned |
| 3 | RW | 0x | bank0 restart  0 = normal operation, 1= restart bank counter to bank0 for next frame write |
| [2:0] | RW | 0x | max bank [2:0]  highest bank\*\_base to use for frame writes before recycling to bank0\_base, only 0 to 4 are valid |

Address: Operational Base + offset 0x00320

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000001000 | Line\_offset [31:0]  Indicates the offset in bytes from the start of one line to the next line.  This value should be equal to or larger than one line of image data and should be word-aligned |

Address: Operational Base + offset 0x00324

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:19] | RO | -- | Line\_offset [31:0]  Indicates the offset in bytes from the start of one line to the next line.  This value should be equal to or larger than one line of image data and should be word-aligned |
| 18 | RO | -- | wbank stop  1 = High pulse at end of frame write to bank |
| 17 | RO |  | wbank start  1 = High pulse at start of frame write to bank |
| 16 | RO | -- | wbank active  1 = wbank\_curr is being written to. Goes high at start of writes, low at last write transfer/completion on axi |
| [15:14] | RO | -- | -- |
| [13:11] | RO | -- | wbank last [2:0]  write bank last active. valid values = 0-4. updated at start of frame write |
| [10:8] | RO | -- | wbank curr [2:0]  write bank currently active. valid values =0-4. updated at start of frame write |
| [7:2] | RO | -- | -- |
| 1 | RO | -- | axi\_port\_enable  enables axi, active high, 1=enables axi write transfers, 0= reset axi domain( via reset synchroniser) |
| 0 | RO | -- | frame write cancel  0 = normal operation, 1= cancel current/future frame write(s), any unstarted AXI bursts cancelled and fifo flushed |

Address: Operational Base + offset 0x00328

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | wbase curr [31:0]  currently active bank base addr - in bytes. updated at start of frame write |

Address: Operational Base + offset 0x0032C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | wbase last [31:0]  last active bank base addr - in bytes. Updated at start of frame write |

Address: Operational Base + offset 0x00330

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RO | -- | frame wcount [15:0]  count of outgoing frame writes (starts) from vdma\_writer sent to AXI output, non resetable, rolls over, updates at pixel 1 of new frame on video in |
| [15:0] | RO | -- | frame icount [15:0]  count of incomming frames (starts) to vdma\_writer on video input, non resetable, rolls over, updates at pixel 1 of new frame on video in |

Address: Operational Base + offset 0x00334

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x000001000 | wxact\_ostand\_limit [7:0]  number oustsanding write transactions(bursts)(responses..1 per burst) limit to raise axi\_fail\_wxact\_ostand. zero disables alarm raise |
| [23:16] | RW | 0x00 | wmaxwait\_limit [7:0]  wvalid maxwait limit(cycles) to raise axi\_fail\_wmaxwait alarm . zero disables alarm raise |
| [15:8] | RW | 0x 01 | awmaxwait\_limit [7:0]  awvalid maxwait limit(cycles) to raise axi\_fail\_awmaxwait alarm . zero disables alarm raise. |
| [7:4] | RW | 0x00 | -- |
| 3 | RW | 0x00 | write timeout disable  At end of frame an optional timeout is applied to wait for AXI writes to completed/accepted befotre caneclling and flushing.  0= Timeout Enabled, timeout count can decrement.  1 = Disable timeout, timeout count can't decrement. |
| 2 | RW | 0x00 | max\_burst\_length\_is\_4  1= Reduce default AXI max\_burst\_length from 16 to 4, 0= Dont reduce( has priority overmax\_burst\_length\_is\_8!) |
| 1 | RW | 0x00 | max\_burst\_length\_is\_8  1= Reduce default AXI max\_burst\_length from 16 to 8, 0= Dont reduce |
| 0 | RW | 0x00 | clear alarms  0>1 transition(synchronous detection) causes local axi/video alarm clear |

Address: Operational Base + offset 0x00338

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:10] | RO | -- | -- |
| 9 | RO | -- | axi\_alarm  active high, problem found on axi port(s)( bresp or awmaxwait or wmaxwait or wxact\_ostand failure ) |
| 8 | RO | -- | video\_alarm  active high, problem found on video port(s) ( active width/height or interline/frame blanks failure). |
| 7 | RO | -- | vi\_fail\_interframe\_blanks  clearable alarm, high to indicate interframe blanking below min |
| 6 | RO | -- | vi\_fail\_interline\_blanks  clearable alarm, high to indicate interline blanking below min |
| 5 | RO | -- | vi\_fail\_active\_height  clearable alarm, high to indicate mismatched active\_height detected ( also raised on missing field!) |
| 4 | RO | -- | vi\_fail\_active\_width  clearable alarm, high to indicate mismatched active\_width detected |
| 3 | RO | -- | axi\_fail\_wxact\_ostand  clearable alarm, high when wxact\_ostand\_limit reached |
| 2 | RO | -- | axi\_fail\_wmaxwait  clearable alarm, high when wmaxwait\_limit reached |
| 1 | RO | -- | axi\_fail\_awmaxwait  clearable alarm, high when awmaxwait\_limit reached |
| 0 | RO | -- | axi\_fail\_bresp  clearable alarm, high to indicate bad bresp captured |

Address: Operational Base + offset 0x0033C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000001000 | blk\_config [31:0]  block configuration (reserved) |

Address: Operational Base + offset 0x00340

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | blk\_status [31:0]  block status output (reserved) |

Zones

Address: Operational Base + offset 0x00400

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | AEXP Weight(0,3) |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | AEXP Weight(0,2) |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | AEXP Weight(0,1) |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AEXP Weight(0,0) |

Address: Operational Base + offset 0x00404

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | AEXP Weight(0,7) |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | AEXP Weight(0,6) |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | AEXP Weight(0,5) |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AEXP Weight(0,4) |

... repeated until ...

Address: Operational Base + offset 0x004DC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | AEXP Weight(14,13) |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | AEXP Weight(14,12) |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | AEXP Weight(14,11) |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AEXP Weight(14,10) |

Address: Operational Base + offset 0x004E0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | -- |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | -- |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | -- |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AEXP Weight(14,14) |

Note: This is an array of 225 4-bit registers, arranged as 15x15. The address for element (i,j) is: 0x00400 + 4 \* int((15i + j) /4)

Address: Operational Base + offset 0x00500

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | AWB Weight (0,3) |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | AWB Weight (0,2) |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | AWB Weight (0,1) |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AWB Weight (0,0) |

Address: Operational Base + offset 0x00504

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | AWB Weight (0,7) |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | AWB Weight (0,6) |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | AWB Weight (0,5) |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AWB Weight (0,4) |

Address: Operational Base + offset 0x005DC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | AWB Weight (14,13) |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | AWB Weight (14,12) |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | AWB Weight (14,11) |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AWB Weight (14,10) |

Address: Operational Base + offset 0x005DC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | -- | -- |
| [27:24] | RW | -- | -- |
| [23:20] | RW | -- | -- |
| [19:16] | RW | -- | -- |
| [15:12] | RW | -- | -- |
| [11:8] | RW | -- | -- |
| [7:4] | RW | -- | -- |
| [3:0] | RW | -- | AWB Weight (14,14) |

Note: This is an array of 225 4-bit registers, arranged as 15x15. The address for element (i,j) is: 0x00500 + 4 \* int((15i + j) / 4)

Video test generator

Address: Operational Base + offset 0x00600

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0xFFFF0300 | r\_backgnd [15:0]  Red background value 16bit, MSB aligned to used width |
| [15:8] | RW | 0x03 | pattern\_type [7:0]  Pattern type select: 0=Flat field,1=Horizontal gradient,2=Vertical Gradient,3=Vertical Bars,4=Rectangle,5-255=Default white frame on black |
| [7:3] | RW | 0x0 | -- |
| 2 | RW | 0x0 | bayer\_rgb\_o\_sel  Bayer or rgb select for output video: 0=bayer, 1=rgb |
| 1 | RW | 0x0 | bayer\_rgb\_i\_sel  Bayer or rgb select for input video: 0=bayer, 1=rgb |
| 0 | RW | 0x0 | test\_pattern\_off\_on  Test pattern off-on: 0=off, 1=on |

Address: Operational Base + offset 0x00604

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0xFFFF | b\_backgnd [15:0]  Blue background value 16bit, MSB aligned to used width |
| [15:0] | RW | 0xFFFF | g\_backgnd [15:0]  Green background value 16bit, MSB aligned to used width |

Address: Operational Base + offset 0x00608

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x8FFF | g\_foregnd [15:0]  Green foreground value 16bit, MSB aligned to used width |
| [15:0] | RW | 0x8FFF | r\_foregnd [15:0]  Red foreground value 16bit, MSB aligned to used width |

Address: Operational Base + offset 0x0060C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x3CAA | rgb\_gradient [15:0]  RGB gradient increment per pixel (0-15.999) Format u4.12  Format: unsigned 4.12-bit fixed-point |
| [15:0] | RW | 0x 8FFF | b\_foregnd [15:0]  Blue foreground value 16bit, MSB aligned to used width |

Address: Operational Base + offset 0x00610

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | 0x00 | -- |
| [27:16] | RW | 0x 01 | rect\_top [11:0]  Rectangle top line number 1-n |
| [15:0] | RW | 0x0000 | rgb\_gradient\_start [15:0]  RGB gradient start value 16bit, MSB aligned to used width |

Address: Operational Base + offset 0x00614

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | 0x0 | -- |
| [27:16] | RW | 0x 001 | rect\_left [11:0]  Rectangle left pixel number 1-n |
| [15:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x100 | rect\_bot [11:0]  Rectangle bottom line number 1-n |

Address: Operational Base + offset 0x00618

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0001 | -- |
| [11:0] | RW | 0x0000 | rect\_right [11:0]  Rectangle right pixel number 1-n |

* + 1. AVIF (IF\_CONFIG)

top

Address: Operational Base + offset 0x00800

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:11] | RW | 0x00000 | -- |
| [10:0] | RW | 0x780 | Active Width [10:0]  Active video width in pixels |

Address: Operational Base + offset 0x00804

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:11] | RW | 0x00000 | -- |
| [10:0] | RW | 0x438 | Active Height [10:0]  Active video height in lines |

Address: Operational Base + offset 0x00808

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:2] | RW | 0x00 | -- |
| [1:0] | RW | 0x0 | AV Data 2 [1:0]  Used to select video stream input to the analytics engine  0 = Select data from Sensor Interface.  1 = Disable (0's)  2 = Select RAW data from DMA Reader.  3 = Select RGB data from DMA Reader |

Address: Operational Base + offset 0x0080C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| 0 | RW | 0x0 | Global FSM reset  1= synchronous reset of FSMs in design (faster recovery after broken frame) |

Sync Generator

Address: Operational Base + offset 0x00820

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| 0 | RW | 0x0 | Generate Mode  frame generation mode: 0 = One Shot, 1 = Continuous |

Address: Operational Base + offset 0x00824

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:22] | RW | 0x0000 | -- |
| [21:0] | RW | 0x0780 | pre fetch dly [21:0]  Delay (in pixels) from frame trigger to first active pixel request. Should be large enough to allow DMA reader FIFO fill sufficiently. |

Address: Operational Base + offset 0x00828

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:22] | RW | 0x000 | -- |
| [21:0] | RW | 0x0780 | interline dly [21:0]  Delay (in pixels) between last active pixel of one line and the first active pixel of the next line. (horizontal blanking) |

Address: Operational Base + offset 0x0082C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:22] | RW | 0x0000 | -- |
| [21:0] | RW | 0x0780 | field dly [21:0]  Delay (in pixels) between last active pixel of last line and the field pulse (end of frame marker). In continous generation mode this will also generate trigger for next frame read |

Address: Operational Base + offset 0x00830

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| 0 | RW | 0x0 | frame\_request  A 0->1 transition triggers a frame read operation. All configuration (for the generator and the DMA reader) must be valid at rising edge |

DMA Reader

Address: Operational Base + offset 0x00840

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:6] | RW | 0x0 | Plane select [1:0]  Plane select for planar base modes. Only used if planar outputs required. Not used. Should be set to 0 |
| [5:4] | RW | 0x0 | -- |
| [3:0] | RW | 0x0 | Base mode [3:0]  Base DMA packing mode for RGB/RAW/YUV etc (see ISP guide) |

Address: Operational Base + offset 0x00848

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:5] | RW | 0x0 | -- |
| 4 | RW | 0x0 | rbase load sel  Selector for rbase\_load strobe: 0-field, 1-configuration bit rbase\_load |
| [3:1] | RW | 0x0 | -- |
| 0 | RW | 0x0 | rbase load |

Address: Operational Base + offset 0x0084C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | rbase [31:0]  Base address for frame buffer, should be word-aligned |

Address: Operational Base + offset 0x00850

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | Line offset [31:0]  Indicates offset in bytes from the start of one line to the next line. Should be word-aligned |

Address: Operational Base + offset 0x00844

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| 0 |  | 0x0 | axi port enable |

Address: Operational Base + offset 0x0085C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:6] | RO | -- | -- |
| 5 | RO | -- | rbank reads on  High inline with rbank\_active for non-null frame reads. Low for null read frames |
| 4 | RO | -- | rbank stop  High for rbank\_start\_stop\_width cyles at end of frame read. Rising edge in line with rbank\_active going low |
| 3 | RO | -- | rbank start  High for rbank\_start\_stop\_width cyles at start of frame read . Rising edge in line with rbank\_active going high |
| 2 | RO | -- | rbank active  1 = rbase bank is active. Goes high at start of reads before frame start, low at frame end (all AXI reads done) |
| 1 | RO | -- | axi alarm  active high, problem found on axi port(s)( rresp or armaxwait or rxnfr\_ostand failure ) |
| 0 | RO | -- | video alarm  active high, problem found on video port(s) ( active width/height or interline/frame blanks failure) |

Address: Operational Base + offset 0x00854

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | CONFIG |

Address: Operational Base + offset 0x00858

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | STATUS |

* + 1. GAMMA\_FE0\_MEM (GAMMA\_FE0\_MEM)

Array

Address: Operational Base + offset 0x02000

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(0)[31:0] |

Address: Operational Base + offset 0x02004

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(1)[31:0] |

... repeated until ...

Address: Operational Base + offset 0x02E7C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(927)[31:0] |

Note: Please see other documentation for a description of the contents of this array.

This is an array of 928 32-bit registers. The address for element (i) is: 0x02000 + 4i

* + 1. HISTOGRAM\_MEM (HISTOGRAM\_MEM)

Array

Address: Operational Base + offset 0x03000

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(0)[31:0] |

Address: Operational Base + offset 0x03004

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(1)[31:0] |

... repeated until ...

Address: Operational Base + offset 0x033FC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(255)[31:0] |

Note: Data [31:0 Please see other documentation for a description of the contents of this array.

This is an array of 256 32-bit registers. The address for element (i) is: 0x03000 + 4i.

* + 1. CMD\_QUEUES (CMD\_QUEUES)

Array

Address: Operational Base + offset 0x03800

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(0)[31:0] |

Address: Operational Base + offset 0x03804

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(1)[31:0] |

... repeated until ...

Address: Operational Base + offset 0x03BFC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | Data(255)[31:0] |

Note: Data [31:0] Please see other documentation for a description of the contents of this array.

This is an array of 256 32-bit registers. The address for element (i) is: 0x03800 + 4i.

* + 1. AV\_CORE (IP\_CONFIG\_ANALYTIX)

HOG

The HOG core computes histogram of oriented gradients and stores it in the memory.

This information can be subsequently used for various analytics projects (recognition, tracking, etc.)

Frame line length after SC96

Frame height after SC96

Address: Operational Base + offset 0x0085C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RO | -- | -- |
| [23:16] | RO | -- | data\_format :  0x01 = RGB32  0x04 = RGB24  0x07 = RAW12  0x0A = YUYV422  0x0B = UYVY422 |
| [15:10] | RO | -- | 0 = R Gr  Gb B    1 = Gr R  B Gb    2 = Gb B  R Gr    3 = B Gb  Gr R |
| [9:8] | RO | -- | rggb\_start :  Starting color of the rggb pattern |
| [7:5] | RO | -- | -- |
| [4:0] | RO | -- | dbg\_select [4:0] :  Enables outputing various debug information for tuning/verification |

Address: Operational Base + offset 0x04200

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | -- | np lut(3) |
| [23:16] | RW | -- | np lut(2) |
| [15:8] | RW | -- | np lut(1) |
| [7:0] | RW | -- | np lut(0) |

Address: Operational Base + offset 0x04204

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | -- | np lut(7) |
| [23:16] | RW | -- | np lut(6) |
| [15:8] | RW | -- | np lut(5) |
| [7:0] | RW | -- | np lut(4) |

... repeated until ...

Address: Operational Base + offset 0x0427C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | -- | np lut(127) |
| [23:16] | RW | -- | np lut(126) |
| [15:8] | RW | -- | np lut(125) |
| [7:0] | RW | -- | np lut(124) |

Note: np\_lut [7:0] Noise profile LUT for Gabor filtering

This is an array of 128 8-bit registers. The address for element (i) is: 0x04200 + 4 \* int(i / 4)

Address: Operational Base + offset 0x04400

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | **--** |
| [11:0] | RW | -- | gab lut(0) |

Address: Operational Base + offset 0x04404

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | **--** |
| [11:0] | RW | -- | gab lut(1) |

... repeated until ...

Address: Operational Base + offset 0x04440

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | **--** |
| [11:0] | RW | -- | gab lut(16) |

Note: gab\_lut [11:0] LUT for Gabor filtering profile LUT for Gabor filtering

This is an array of 17 12-bit registers. The address for element (i) is: 0x04400 + 4i

Address: Operational Base + offset 0x004300

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:25] | RO | 0x0 | -- |
| 24 | RO | 0x0 | data\_format [7:0]  0x01 = RGB32  0x04 = RGB24  0x07 = RAW12  0x0A = YUYV422  0x0B = UYVY422 |
| [15:8] | RO | 0x0 | 0 = R Gr  Gb B    1 = Gr R  B Gb    2 = Gb B  R Gr    3 = B Gb  Gr R |
| [7:0] | RO | 0x0 | rggb\_start [1:0]  Starting color of the rggb pattern |

Address: Operational Base + offset 0x04304

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | 0x0 | -- |
| 24 | RW | 0x0 | uu\_thresh [11:0]  Threshold for the range of undirectional detection in 0.12 format |
| [15:12] | RW | 0xB | da\_slope [7:0]  Direction priority slope |
| [11:0] | RW | 0x69A | uu\_slope [7:0]  Undirectional slope |

Address: Operational Base + offset 0x004308

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:28] | RW | 0x0 | -- |
| [27:16] | RW | 0x0 | uu\_offset [11:0]  Offset for the range of undirectional detection in 0.12 format |
| [15:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | da\_thresh [11:0]  Threshold for the range of direction priority detection in 0.12 format |

Address: Operational Base + offset 0x00430C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0 | -- |
| [11:0] | RW | 0x0 | da\_offset [11:0]  Offset for the range of direction priority detection in 0.12 format |

Address: Operational Base + offset 0x004310

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0000 | -- |
| [15:8] | RW | 0x66 | sens\_dg\_ks7 [7:0]  Sensitivity of diagonal edges for kernel size 7. Inverse (1/s) |
| [7:0] | RW | 0x6F | sens\_vh\_ks7 [7:0]  Sensitivity of vertical/horizontal edges for kernel size 7. Inverse (1/s) |

Address: Operational Base + offset 0x004318

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB77000 | C BASEADDR sc96 off0 |

Address: Operational Base + offset 0x00431C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB775A0 | C BASEADDR sc48 off0 |

Address: Operational Base + offset 0x004320

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB78E90 | C BASEADDR sc24 off0 |

Address: Operational Base + offset 0x004324

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB7F9D0 | C BASEADDR sc12 off0 |

Address: Operational Base + offset 0x004328

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB9B140 | C BASEADDR sc80 off0 |

Address: Operational Base + offset 0x00432C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB9B980 | C BASEADDR sc40 off0 |

Address: Operational Base + offset 0x004330

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CB9DEE0 | C BASEADDR sc20 off0 |

Address: Operational Base + offset 0x004334

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CBA7A90 | C BASEADDR sc10 off0 |

Address: Operational Base + offset 0x004338

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CBCF5E0 | C BASEADDR sc72 off0 |

Address: Operational Base + offset 0x00433C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CBD00D0 | C BASEADDR sc36 off0 |

Address: Operational Base + offset 0x004340

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CBD2F08 | C BASEADDR sc18 off0 |

Address: Operational Base + offset 0x004344

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CBDF0A0 | C BASEADDR sc09 off0 |

Address: Operational Base + offset 0x004348

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC10148 | C BASEADDR sc64 off0 |

Address: Operational Base + offset 0x00434C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC10E68 | C BASEADDR sc32 off0 |

Address: Operational Base + offset 0x004350

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC14868 | C BASEADDR sc16 off0 |

Address: Operational Base + offset 0x004354

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC23BC8 | C BASEADDR sc08 off0 |

Address: Operational Base + offset 0x004358

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC62068 | C BASEADDR sc56 off0 |

Address: Operational Base + offset 0x00435C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC63268 | C BASEADDR sc28 off0 |

Address: Operational Base + offset 0x004360

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC67FE0 | C BASEADDR sc14 off0 |

Address: Operational Base + offset 0x004364

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x2CC7C080 | C BASEADDR sc07 off0 |

Address: Operational Base + offset 0x004368

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x0001B770 | C FDSIZE sc12 off0 |

Address: Operational Base + offset 0x00436C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00006B40 | C FDSIZE sc24 off0 |

Address: Operational Base + offset 0x004370

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000018F0 | C FDSIZE sc48 off0 |

Address: Operational Base + offset 0x004374

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000005A0 | C FDSIZE sc96 off0 |

Address: Operational Base + offset 0x004378

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00027B50 | C FDSIZE sc10 off0 |

Address: Operational Base + offset 0x00437C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00009BB0 | C FDSIZE sc20 off0 |

Address: Operational Base + offset 0x004380

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00002560 | C FDSIZE sc40 off0 |

Address: Operational Base + offset 0x004384

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000840 | C FDSIZE sc80 off0 |

Address: Operational Base + offset 0x004388

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000310A8 | C FDSIZE sc09 off0 |

Address: Operational Base + offset 0x00438C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x0000C198 | C FDSIZE sc18 off0 |

Address: Operational Base + offset 0x004390

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00002E38 | C FDSIZE sc36 |

Address: Operational Base + offset 0x004394

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000AF0 | C FDSIZE sc72 off0 |

Address: Operational Base + offset 0x004398

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x0003E4A0 | C FDSIZE sc08 off0 |

Address: Operational Base + offset 0x00439C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x0000F360 | C FDSIZE sc16 off0 |

Address: 0x0043A0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00003A00 | C FDSIZE sc64 off0 |

Address: Operational Base + offset 0x0043A4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000D20 | C FDSIZE sc32 off0 |

Address: Operational Base + offset 0x0043A8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00051480 | C FDSIZE sc64 off0 |

Address: Operational Base + offset 0x0043AC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x000140A0 | C FDSIZE sc14 off0 |

Address: Operational Base + offset 0x0043B0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00004D78 | C FDSIZE sc28 off0 |

Address: Operational Base + offset 0x0043B4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00001200 | C FDSIZE sc56 off0 |

Address: Operational Base + offset 0x004454

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:20] | RW | 0x0 | -- |
| [19:0] | RW | 0x0 | dma\_enable [19:0]  Enables writing hog data to DMA |

Address: Operational Base + offset 0x004458

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RW | 0x0 | -- |
| [3:0] | RW | 0x0 | offset [3:0]  HOG Offset |

Address: Operational Base + offset 0x0044F4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:3] | RW | 0x0 | -- |
| 2 | RW | 0x0 | rst\_fsm  bit 2: Delay-line FSM(active high): |
| [1:0] | RW | 0x3 | rstn\_int [1:0]  Soft reset (active low):  bit 0: Video;  bit 1: AXI; |

Address: Operational Base + offset 0x0044F8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RO | -- | -- |
| [7:0] | RO | -- | time\_stamp [7:0]  Time stamp for HOG data |

Address: Operational Base + offset 0x004900

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | -- | -- |
| 0 | RW | -- | dump\_enable  Enables or disables hog dump in the model |

Address: Operational Base + offset 0x004460

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | C BASEADDRC BASEADDR SVM AXI WRITTER |

Address: Operational Base + offset 0x004464

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | C BASEADDR SVM AXI READER |

Address: Operational Base + offset 0x004468

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | -- |
| [11:0] | RW | -- | line length |

Address: Operational Base + offset 0x00446C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | -- | -- |
| [11:0] | RW | -- | lines number |

Address: Operational Base + offset 0x004470

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:11] | RW | -- | -- |
| [10:0] | RW | -- | svm norm offset |

Address: Operational Base + offset 0x004474

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | model program |

Address: Operational Base + offset 0x004478

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | -- | -- |
| [7:0] | RW | -- | model start |

Address: Operational Base + offset 0x00447C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | -- | -- |
| [7:0] | RW | -- | model end |

Address: Operational Base + offset 0x004480

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | -- | -- |
| [7:0] | RW | -- | model x |

Address: Operational Base + offset 0x004484

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | -- | -- |
| [7:0] | RW | -- | model y |

Address: Operational Base + offset 0x004488

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | -- | -- |
| 0 | RW | -- | Data start |

Address: Operational Base + offset 0x00448C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | -- | -- |
| 0 | RW | -- | Data end |

Address: Operational Base + offset 0x004490

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:26] | RW | -- | -- |
| [25:16] | RW | -- | data in pedestal |
| [15:10] | RW | -- | -- |
| [9:0] | RW | -- | data in mult |

Address: Operational Base + offset 0x004494

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:5] | RW | -- | -- |
| [4:0] | RW | -- | data in scale |

Address: Operational Base + offset 0x004498

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | -- | -- |
| [7:0] | RW | -- | data in offset |

Address: Operational Base + offset 0x00449C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | -- | -- |
| 0 | RW | -- | dma enable wr pm |

Address: Operational Base + offset 0x0044A0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | -- | dma\_data\_sent [31:0]  Data sent indicator from SVM writter |

Address: Operational Base + offset 0x004500

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000090 | -- |
| [11:0] | RW |  | C LENGTH sc96 |

Address: Operational Base + offset 0x004504

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x0130 | C LENGTH sc48 |

Address: Operational Base + offset 0x004508

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x270 | **C LENGTH sc24** |

Address: Operational Base + offset 0x00450C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x000 | -- |
| [11:0] | RW | 0x4F0 | C LENGTH sc12 |

Address: Operational Base + offset 0x004510

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x0B0 | C LENGTH sc80 |

Address: Operational Base + offset 0x004514

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x170 | C LENGTH sc40 |

Address: Operational Base + offset 0x004518

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x000002F0 | -- |
| [11:0] | RW | 0x2F0 | C LENGTH sc20 |

Address: Operational Base + offset 0x00451C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x5F0 | C LENGTH sc10 |

Address: Operational Base + offset 0x004520

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x0C8 | C LENGTH sc72 |

Address: Operational Base + offset 0x004524

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x198 | C LENGTH sc36 |

Address: Operational Base + offset 0x004528

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x348 | C LENGTH sc18 |

Address: Operational Base + offset 0x00452C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x698 | C LENGTH sc09 |

Address: Operational Base + offset 0x004530

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x0E0 | C LENGTH sc64 |

Address: Operational Base + offset 0x004534

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x1D0 | C LENGTH sc32 |

Address: Operational Base + offset 0x004538

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x3B0 | C LENGTH sc16 |

Address: Operational Base + offset 0x00453C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x770 | C LENGTH sc08 |

Address: Operational Base + offset 0x004540

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x100 | C LENGTH sc56 |

Address: Operational Base + offset 0x004544

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x218 | C LENGTH sc28 |

Address: Operational Base + offset 0x004548

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x438 | C LENGTH sc14 |

Address: Operational Base + offset 0x00454C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x880 | C LNUMBER sc07 |

Address: Operational Base + offset 0x004550

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x009 | C LNUMBER sc96 |

Address: Operational Base + offset 0x004554

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x014 | C LNUMBER sc48 |

Address: Operational Base + offset 0x004558

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x270 | C LNUMBER sc24 |

Address: Operational Base + offset 0x00455C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x058 | C LNUMBER sc12 |

Address: Operational Base + offset 0x004560

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0000000B | -- |
| [11:0] | RW | 0x00B | C LNUMBER sc80 |

Address: Operational Base + offset 0x004564

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x019 | C LNUMBER sc40 |

Address: Operational Base + offset 0x004568

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x034 | C LNUMBER sc20 |

Address: Operational Base + offset 0x00456C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x00D | C LNUMBER sc10 |

Address: Operational Base + offset 0x004570

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x01C | C LNUMBER sc072 |

Address: Operational Base + offset 0x004574

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x03A | C LNUMBER sc36 |

Address: Operational Base + offset 0x004578

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000076 | -- |
| [11:0] | RW | 0x076 | C LNUMBER sc18 |

Address: Operational Base + offset 0x00457C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x 00E | C LNUMBER sc09 |

Address: Operational Base + offset 0x004580

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x0000001F | -- |
| [11:0] | RW |  | C LNUMBER sc64 |

Address: Operational Base + offset 0x004584

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000041 | -- |
| [11:0] | RW | 0x041 | C LNUMBER sc32 |

Address: Operational Base + offset 0x00458C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x085 | C LNUMBER sc16 |

Address: Operational Base + offset 0x004590

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x011 | C LNUMBER sc08 |

Address: Operational Base + offset 0x004594

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x024 | C LNUMBER sc56 |

Address: Operational Base + offset 0x004598

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x04B | C LNUMBER sc28 |

Address: Operational Base + offset 0x00459C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:12] | RW | 0x00000 | -- |
| [11:0] | RW | 0x098 | C LNUMBER sc14 |

Address: Operational Base + offset 0x004700

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0000 | -- |
| [15:0] | RW | 0x0098 | C SVM MODEL ENABLE |

dbg

Address: Operational Base + offset 0x0044A4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:10] | RO | -- | NA1 [21:0]  NA1 |
| [9:0] | RO | -- | Norm [9:0]  Data normalization value output |

Address: Operational Base + offset 0x0044A8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | -- | '0' when normalization process is active |
| 30 | RO | -- | '1' when data read from first cache into normalization happens |
| [29:0] | RO | -- | NA2 [29:0]  NA2 |

Address: Operational Base + offset 0x0044AC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x0003E4A0 | cp3 [31:0]  Constant value X"89ABCDEF" |

Address: Operational Base + offset 0x0044B0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x0003E4A0 | cp4 [31:0]  Constant value X"98765432" |

Address: Operational Base + offset 0x0044E0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RO | -- | HOG ID |
| [15:0] | RO | -- | Frame Offset |

* + 1. CONTROLLER

SVM/HOG controller block

Address: Operational Base + offset 0x004600

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| 0 | RW | 0x0 | CONTROLLER Reset  '0' - reset inactive  '1' - controller is held in reset  Note: this is not auto-cleared |

Address: Operational Base + offset 0x004604

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| 0 | RW | 0x0 | ENABLE  ENABLE[0] - '0' - controller disabled  ENABLE[0] - '1' - controller enabled  Note: set up all config and cycle reset before enabling this bit |

Address: Operational Base + offset 0x004608

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RO | 0x0 | -- |
| [3:0] | RO | 0xF | OFFSET control [3:0]  Offset selection  '0000' - no offsets selected  '0001' - offset 1 selected  '0011' - offset 1,2 selected  '0111' - offset 1,2,3 selected  '1111' - offset 1,2,3,4 selected |

Address: Operational Base + offset 0x00460C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | SVM Detection count [31:0]  SVM detection count  Contains number of detections per super-frame |

Address: Operational Base + offset 0x004610

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | Error stat register [31:0]  Controller alarm indicators  ERROR[19:0] - indicates error with config setup  ERROR[20] - HOG overrun error  ERROR[21] - SVM overrun error |

Address: Operational Base + offset 0x004614

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x00 | -- |
| 0 | RW | 0x0 | CONTROLLER Status clear  Set '1' to clear alarm indicators  Note: this is not auto-cleared |

Address: Operational Base + offset 0x004618

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | -- | -- |
| [15:8] | RW | -- | SVM timestamp [7:0]  Timestamp  Timestamp data(lower 6b given to SVM) |
| [7:1] | RW | -- | -- |
| 0 | RW | -- | SVM buffer valid indicator  Indicates which detection buffer area is valid  BufferValid[0]  '0' - starts at SVM write base address  '1' - starts at SVM write base address + 69K |

Address: Operational Base + offset 0x00461C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:2] | RW | 0x0 | -- |
| 1 | RW | 0x0 | SVMCMP  SVMCMP[1] - '0' - SVM compensation in-active  SVMCMP[1] - '1' - SVM compensation active  Note: SVM comp. cancels HOG transation at start of frame if SVM is currently active |
| 0 | RW | 0x0 | SVMSTP  SVMSTP[1] - '0' - SVM stop in-active  SVMSTP[1] - '1' - SVM stop active  Note: SVM stop cancels any further SVM processing and returns controller SVM state machine to idle |

Address: Operational Base + offset 0x004620

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | -- | -- |
| [15:0] | RW | -- | CONTROLLER status indicator [15:0]  Indicates Controller activity  STAT[4:0] - HOG state  STAT[6] - HOG busy  STAT[7] - SVM busy  STAT[15:8] - SVM state |

Address: Operational Base + offset 0x0046F0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RO | -- | Build Version Register [31:0]  Build Version Register  Version[31:16] - Year  Version[ 15:8] - Month  Version[ 7:0] - Day |

* + 1. CFM (IP\_CONFIG\_CFM)

Address: Operational Base + offset 0x005100

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| [0] | RW | 0x0 | CFM Reset  '0' - reset inactive  '1' - controller is held in reset  Note: this is not auto-cleared |

Address: Operational Base + offset 0x005104

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:1] | RW | 0x0 | -- |
| [0] | RW | 0x0 | Start Proc  START\_PROC[0] - '0' - controller disabled  START\_PROC[0] - '1' - controller enabled  Note: set up all config and cycle reset before enabling this bit  Note: Bring bit high and then low to activate block |

Address: Operational Base + offset 0x005108

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | CFM Cluster 0 base address [31:0]  Cluster config data base address |

Address: Operational Base + offset 0x00510C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x00000000 | -- |
| [7:0] | RW | CNT [7:0]  CNT[7:0] - Number of clusters to process |

Address: Operational Base + offset 0x005110

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | CFM Seed base address [31:0]  Seed data base address |

Address: Operational Base + offset 0x005114

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | -- |
| [15:0] | RW | 0x0 | MASK [15:0]  Seed selection mask  Not used in current build |

Address: Operational Base + offset 0x005118

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | OffsetX 3 X offset controls used in Gaussian convolution |
| [23:16] | RW | 0x0 | OffsetX 2 X offset controls used in Gaussian convolution |
| [15:8] | RW | 0x0 | OffsetX 1 X offset controls used in Gaussian convolution |
| [7:0] | RW | 0x0 | OffsetX 0 X offset controls used in Gaussian convolution |

Address: Operational Base + offset 0x00511C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | **--** |
| [23:16] | RW | 0x0 | OffsetX 6 X offset controls used in Gaussian convolution |
| [15:8] | RW | 0x0 | OffsetX 5 X offset controls used in Gaussian convolution |
| [7:0] | RW | 0x0 | OffsetX 4 X offset controls used in Gaussian convolution |

Address: Operational Base + offset 0x005120

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | OffsetY 3 Yoffset controls used in Gaussian convolution |
| [23:16] | RW | 0x0 | OffsetY 2 Y offset controls used in Gaussian convolution |
| [15:8] | RW | 0x0 | OffsetY 1 Y offset controls used in Gaussian convolution |
| [7:0] | RW | 0x0 | OffsetY 0 Y offset controls used in Gaussian convolution |

Address: Operational Base + offset 0x005124

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | -- |
| [23:16] | RW | 0x0 | OffsetY 6 Y offset controls used in Gaussian convolution |
| [15:8] | RW | 0x0 | OffsetY 5 Y offset controls used in Gaussian convolution |
| [7:0] | RW | 0x0 | OffsetY4 Y offset controls used in Gaussian convolution |

Address: Operational Base + offset 0x005128

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | Coeff X 3 Coefficients used in bi-linear interpolationg filter |
| [23:16] | RW | 0x0 | CoeffX2 Coefficients used in bi-linear interpolationg filter |
| [15:8] | RW | 0x0 | OffsetX1 Coefficients used in bi-linear interpolationg filter |
| [7:0] | RW | 0x0 | OffsetX 0 Coefficients used in bi-linear interpolationg filter |

Address: Operational Base + offset 0x00512C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | -- |
| [23:16] | RW | 0x0 | CoeffX 6 Coefficients used in bi-linear interpolationg filter |
| [15:8] | RW | 0x0 | CoeffX 5 Coefficients used in bi-linear interpolationg filter |
| [7:0] | RW | 0x0 | CoeffX 4 Coefficients used in bi-linear interpolationg filter |

Address: Operational Base + offset 0x005138

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | CoeffY 3 Coefficients used in bi-linear interpolationg filter |
| [23:16] | RW | 0x0 | CoeffY2 Coefficients used in bi-linear interpolationg filter |
| [15:8] | RW | 0x0 | OffsetY 1 Coefficients used in bi-linear interpolationg filter |
| [7:0] | RW | 0x0 | OffsetY 0 Coefficients used in bi-linear interpolationg filter |

Address: Operational Base + offset 0x00513C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:24] | RW | 0x0 | -- |
| [23:16] | RW | 0x0 | CoeffY 6 Coefficients used in bi-linear interpolationg filter |
| [15:8] | RW | 0x0 | CoeffY 5 Coefficients used in bi-linear interpolationg filter |
| [7:0] | RW | 0x0 | CoeffY 4 Coefficients used in bi-linear interpolationg filter |

Address: Operational Base + offset 0x005148

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | AXI Config transfer size in bytes  Num Thumbnails : Size  1 : 14 x 4 (0x38)  2 : 20 x 4 (0x50)  3 : 28 x 4 (0x70)  4 : 34 x 4 (0x88) |
| [15:0] | RW | 0x0 | AXI Config transfer size in bytes  Num Thumbnails : Size  1 : 14 x 4 (0x38)  2 : 20 x 4 (0x50)  3 : 28 x 4 (0x70)  4 : 34 x 4 (0x88) |

Address: Operational Base + offset0x0514c

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:10] | RW | 0x0 | -- |
| [9:8] | RW | 0x0 | plane\_select [1:0]  plane\_select[1:0]  (YUV input only - not used for current build) |
| [7:0] |  | 0x0 | format [7:0]  0x1: RGB32  0x4: RGB24  0x7: RAW12  0xA: YUYV422  0xB: UYVY422 |

Address: Operational Base + offset 0x005150

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RW | 0x0 | -- |
| [3:0] | RW | 0x0 | Alpha Factor [3:0]  Alpha sum factor used in Thumbnail mean/variance generation |

Address: Operational Base + offset 0x005154

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RW | 0x0 | -- |
| [3:0] | RW | 0x0 | Alpha Factor [3:0]  Alpha sum factor used in Thumbnail mean/variance generation |

Address: Operational Base + offset 0x005158

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RW | 0x0 | -- |
| [3:0] | RW | 0x0 | Beta Factor [3:0]  Beta sum factor used in Thumbnail mean/variance generation |

Address: Operational Base + offset 0x00515C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DOTP DMA write address [31:0]  Write data base address |

Address: Operational Base + offset 0x005160

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | Model data base address |

Address: Operational Base + offset 0x005164

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:30] | RW | 0x0 | -- |
| [29:20] | RW | 0x0 | SEG Y ADDON [9:0] Multipy factor used in segmentation |
| [19:16] | RW | 0x0 | SEG Y ADDON [9:0] Multipy factor used in segmentation |
| [15:10] | RW | 0x0 | SEG Y ADDON [9:0] Multipy factor used in segmentation |
| [9:0] | RW | 0x0 | SEG Y ADDON [9:0] Multipy factor used in segmentation |

Address: Operational Base + offset 0x005168

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:5] | RW | 0x0 | -- |
| [4:0] | RW | 0x0 | SEG Factor [4:0]  Multipy factor used in segmentation |

Address: Operational Base + offset 0x00516C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:8] | RW | 0x0 | -- |
| [7:0] | RW | 0x0 | SEG Offset [7:0]  Multipy factor used in segmentation |

Address: Operational Base + offset 0x005170

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:4] | RW | 0x0 | -- |
| [3:0] | RW | 0x0 | SEG Norm factor [3:0]  Normalization factor used in segmentation |

Address: Operational Base + offset 0x005174

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | Pixel Image base address [31:0]  Cluster Pixel data base address |

Address: Operational Base + offset 0x005178

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | -- |
| [15:0] | RW | 0x0 | Error status [15:0]  Error status[ 0] : AXI D'load error  Error status[ 1] : Cluster config error  Error status[ 2] : Zero Cluster cnt selected  Error status[ 3] : RD FIFO Full and write  Error status[ 4] : RD FIFO Empty and read  Error status[ 5] : WR FIFO Full and write  Error status[ 6] : WR FIFO Empty and read  Error status[ 7] : AXI transfer error  Error status[ 8] : Unpacker underflow error  Error status[ 9] : Thumbnail 0 extraction error  Error status[10] : AXI unaligned read address error  Error status[11] : AXI unaligned write address error  Error status[15:12] : reserved |

Address: Operational Base + offset 0x00517C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | CFM status [31:0]  Status[15:0]  0x0001: CFM IDLE  0x0004: CFM d'load cluster config  0x0010: CFM d'load seed data  0x0040: CFM d'load image data  0x0100: CFM d'load model data  Status[31:16]  Status[16]: Unpacker input field ok  Status[17]: Unpacker output field ok  Status[18]: RD FIFO FILL level ok  Status[19]: Thumbnail Extracted ok  Status[20]: SEG processing done ok  Status[21]: DOTP processing done ok  Status[22]: All DOTP cluster processing done ok  Status[31:23]: reserved |

Address: Operational Base + offset 0x005180

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Cluster Pixel Offset Location [31:0]  he offset from Cluster Pixel base address to the start of the cluster |

Address: Operational Base + offset 0x005184

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Cluster width (in bytes) [31:0]  The cluster width in bytes (word aligned) |

Address: Operational Base + offset 0x005188

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Cluster width (in pixels) [31:0]  The cluster width in pixels |

Address: Operational Base + offset 0x00518C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Cluster height [31:0]  The cluster height |

Address: Operational Base + offset 0x005190

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:2] | RW | 0x0 | -- |
| [1:0] | RW | 0x0 | DEBUG: CFM RGGB start pos [1:0]  The RGGB starting position |

Address: Operational Base + offset 0x005194

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:16] | RW | 0x0 | -- |
| [15:0] | RW | 0x0 | DEBUG: CFM Cluster image line stride [15:0]  The Cluster image stride width (in bytes) |

Address: Operational Base + offset 0x005198

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Thm nail 0 X start [31:0]  Thm nail 0 X start position |

Address: Operational Base + offset 0x00519C

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Thm nail 0 Y start [31:0]  Thm nail 0 Y start position |

Address: Operational Base + offset 0x0051A0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Thm nail 0 X end [31:0]  Thm nail 0 X end position |

Address: Operational Base + offset 0x0051A4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Thm nail 0 Y end [31:0]  Thm nail 0 Y end position |

Address: Operational Base + offset 0x0051A8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Thm nail X offset [31:0]  Thm nail 0 X offset |

Address: Operational Base + offset 0x0051AC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Thm nail Y offset [31:0]  Thm nail 0 Y offset |

Address: Operational Base + offset 0x0051B0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:5] | RW | 0x00000000 | -- |
| [4:0] | RW | EBUG: CFM Thm nail 0 model number [4:0]  hm nail 0 model number |

Address: Operational Base + offset 0x0051B4

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | DEBUG: CFM Dotp output [31:0]  The CFM detection word |

Address: Operational Base + offset 0x0051B8

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:6] | RW | 0x00000000 | -- |
| [5:0] | RW | DEBUG: CFM Cluster number [5:0]  The Current CFM cluster number |

Address: Operational Base + offset 0x005FFC

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | CFM Build ID register [31:0]  The Current CFM cluster number  [31:24] : Year(less 2K)  [23:16] : Month  [15:08] : Day  [07:00] : Tag ID |

* + 1. INTERRUPT\_CONTROLLER\_WRAPPER

Int\_ctrl

Address: Operational Base + offset 0x006000

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | status [31:0]  Interrupt Status Register. All the interrupt signals are active high  status[31:29]: Reserved (always 0)  status[28]: cfm\_error\_flg\_irq  status[27]: cfm\_tnm\_done\_irq  status[26]: av\_core\_frame\_end\_irq  status[25]: av\_core\_clf\_done\_irq  status[24]: av\_core\_dma\_write\_done\_wr\_pm\_irq  status[23]: av\_core\_dma\_write\_done\_irq  status[22]: av\_core\_axi\_fail\_rresp\_o\_rd\_pm\_irq  status[21]: av\_core\_axi\_fail\_bresp\_o\_wr\_pm\_irq  status[20]: av\_core\_wfifo\_fail\_full\_wr\_pm\_irq  status[19]: av\_core\_hog\_bad\_vbi\_flg\_irq  status[18]: av\_core\_hog\_br\_fr\_flg\_irq  status[17:13]: av\_core\_axi\_fail\_bresp\_irq  status[12:8]: av\_core\_wfifo\_fail\_full\_irq  status[7:0]: av\_si\_interrupt\_request\_irq |

Address: Operational Base + offset 0x006008

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| [31:0] | RW | 0x00000000 | mask [31:0]  Interrupt Mask Register. Used to enable/disable the individual interrupt signals present in the interrupt status register to assert the irq\_level and irq\_pulse output ports  0: Interrupt signal is not masked  1: Interrupt signal is masked |

* 1. Programming Sequence

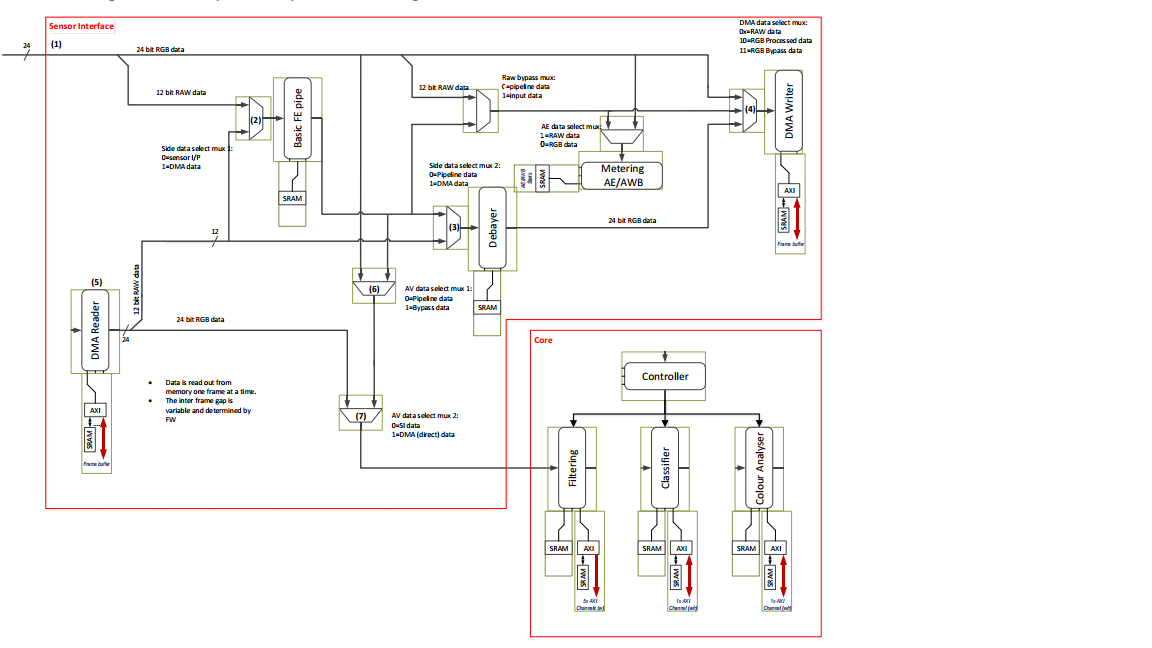


Figure .IVA diagram block

Depending on the application, the Spirit IP enables a wide range of data feeding scenarios into the Core.

With reference to Figure 5 the data flows are listed below:

1. Raw sensor data input via FB (Raw). (1)=> (2) => (4) => (5) => (7): This is a main mode of connecting to the sensor, where the raw sensor data is processed by the SI and sent to the Core via DMA in RGB format.
2. Raw sensor data input via FB (Raw). (1)=> (2) => (3) => (4) => (5) => (7): This is a main mode of connecting to the sensor, where raw sensor data is processed by the SI and sent t o the Core via DMA in RGB format.
3. RGB input stream via FB (RGB) (1) => (4) => (5) => (7): A mode of connecting to the sensor or other source which produces RGB data.
4. Raw sensor data input via FB (Raw).Tiled processing (1) => (4); (5) => (2) => (6) => (7): This is a mode to connect to the sensor which produces images larger than can be stored in the Spirit delay lines. In this case the raw data is fed directly to the DMA Writer (4) and subsequently read and processed as tiles by the SI and Core. While it enables connecting a sensor with higher resolution than enabled by the Spirit delay lines (SRAM), additional bandwidth and processing overheads have to be factored in due to overlapped readout.
5. Raw sensor data input via FBRGB readout for multi-content processing. (1)=> (2) => (3) => (4) ;( 5) => (7).This mode is a variation of 2 and enables processing multiple video sources (e.g. one rom direct input, second from memory).
6. Direct raw sensor data input (no Colour Filtering) (1) => (2) => (6) => (7): This mode supports feeding the sensor data directly into the Core, omitting the FB. In this case an important analytics stage, called Colour Filtering (CF), is skipped, which may result in lower detection performance (higher false positive rate).
7. Direct RGB stream (no Colour Filtering) (1)=>(6)=>(7):This mode supports feeding sensor the data directly into the Core,omitting the FB.In this case an important analytics stage, called Colour Filtering (CF),is skipped,which may result in lower detection performance(higher false positive rate).
8. Memory Raw input (5) => (2) => (6) => (7): This mode is used when video data is input only via the external memory.

Memory RGB input (5) => (7): This mode is used when video data is input only via the external memory.

1. Crypto Engine
   1. Register Summary

The following table outlines the global registers and the memory ranges reserved for the various components of the SPAcc-PDU. The total memory map requires 512 kB of space on the bus (19 bits of address). All undefined ranges are considered reserved. Bit-Fields marked with an asterisk (\*) are either optional, or width configurable. The shown values indicate the maximal configuration.

There are base address in SPAcc,TRNG,PKA sub-blocks which show in SPDU registers.As the address of SPAcc is 0x40000(opreation base)+offset.

Table 344.Crypto Engine Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| SPDU\_IRQ\_EN | 0x0\_0000 | 0x0 | Enables interrupts |
| SPDU\_IRQ\_STAT | 0x0\_0004 | 0x0 | Individual interrupt bits corresponding directly to the bits within the IRQ\_EN register. These are level sensitive and indicate the current state of the interrupts from the corresponding engine. To clear these bits the appropriate acknowledgment must be directed at the individual engine. The global O\_irq signal is asserted if the appropriate enable(s) are set and the corresponding status bits are also set |
| DMA\_BRST\_SZ | 0x0\_0010 | 0x10 | Maximum size of DMA burst over the master bus (in words). Reset value is 16. Minimum setting is 4. A value of zero in this register will set the burst size to maximum (64 words).  This register replaces the SPAcc SDMA\_BRST\_SZ register, which is considered reserved in this design. |
| SP\_DECODE | 0x0\_0014 | 0x0 | Upper bits of master address to decode into the scratchpad memory |
| SPACC\_CMD\_PRIORITY | 0x0\_0018 | 0x0 | Set the SPAcc command priority level that the accelerator engines run at. 0 is highest (runs before CMD0 FIFO), 3 is lowest (runs after CMD2 FIFO). |
| SP\_VERSION | 0x0\_0180 | 0x0604\_8b4b | Copy of the SPAcc version registers. Corresponds to vSPAcc 0 if multiple virtual SPAcc engines are present |
| SP\_VERSION\_EXT | 0x0\_0184 | 0x1572\_000c |
| SPDU\_VERSION | 0x0\_0188 | 0x326 | Flags indicating |
| RE | 0x0\_8000-0x0\_BFFF | N/A | Record Engine component |
| MPM | 0x0\_C000-0x0\_FFFF | N/A | Multi-Packet Manager component |
| KEP | 0x1\_0000-0x1\_3FFF | N/A | Key Exchange Protocol component |
| EA | 0x1\_4000-0x1\_7FFF | N/A | ESP/AH Engine component. |
| RNG | 0x1\_8000-0x1\_FFFF | N/A | Random Number Generator component. |
| PKA | 0x2\_0000-0x3\_FFFF | N/A | Public Key Accelerator component. |
| SPACC | 0x4\_0000-0x7\_FFFF | N/A | SPAcc component. For configurations with SPAcc virtualization turned on, this maps to the virtual SPAcc component at index 0 |
| SPAcc core register | | | |
| IRQ\_EN | 0x0\_0000 | 0x0 | SPAcc interrupt enable |
| IRQ\_STAT | 0x0\_0004 | 0x0 | Individual interrupts. These correspond directly with the associated bit in the IRQ\_EN register. Write 1 to clear. |
| IRQ\_CTRL  (SPAcc-QoS) | 0x0\_0008 | 0x2400\_0000 | Configures the individual command FIFO counts and the status FIFO count |
| IRQ\_CTRL  (SPAcc) | 0x0\_0008 | 0x2400\_0000 | See above for SPAcc-QoS version of register. |
| FIFO\_STAT  (SPAcc-QoS) | 0x0\_000C | 0x8000\_0000 | FIFO state register |
| FIFO\_STAT  (SPAcc) | 0x0\_000C | 0X8000\_0000 | Similar to SPAcc-QoS version of register. (See above) |
| SDMA\_BRST\_SZ | 0x0\_0010 | 0x0 | Maximum size of DMA burst over the master bus (in words). Reset value is 16. Minimum setting is 4.Avalue of zero in this register will set the burst size to maximum (32words). Width configures to only 4bits wide when SHA-384/512 algorithms are not enabled (hence reset value is zero in this case).This register is shared globally between all virtual SPAcc components. |
| SRC\_PTR | 0x0\_0020 | 0x0 | Pointer to source packet data structure in system memory. |
| DST\_PTR | 0x0\_0024 | 0x0 | Pointer to destination packet data structure in system memory. |
| OFFSET | 0x0\_0028 | 0x0 | Offsets in bytes in the packet data structure to start of AAD. |
| PRE\_AAD\_LEN | 0x0\_002C | 0x0 | Byte length of the prefixed AAD |
| POST\_AAD\_LEN | 0x0\_0030 | 0x0 | Byte length of the post-fixed AAD. |
| PROC\_LEN | 0x0\_0034 | 0x0 | Total packet byte length for processing, including PRE\_AAD\_LEN and payload. |
| ICV\_LEN | 0x0\_0038 | 0x0 | Byte length of the ICV. If set to zero, the default length for the selected hash algorithm will be used. |
| ICV\_OFFSET | 0x0\_003C | 0x0 | Byte offset in packet data structure to start of ICV. Refers to SRC on decrypt, and DST on encrypt. This value is ignored if the CTRL.ICV\_APPEND flag is set. |
| IV\_OFFSET | 0x0\_0040 | 0x0 | Byte offset in SRC data structure to obtain the cryptographic initialization value. Note: This register is only present if pIV\_IMPORT\_EN is turned on. |
| SW\_CTRL | 0x0\_0044 | 0x0 | Software tag of packet ID. Will be returned with the packet in the outbound status FIFO. This value automatically increments on every push to the command FIFO. |
| AUX\_INFO | 0x0\_0048 | 0x0 | CRC mode indicating |
| CTRL | 0x0\_004C | 0xc000 | SPAcc control register |
| STAT\_POP | 0x0\_0050 | 0x0 | Writing any value to this register causes a result to be popped off of the status FIFO. Must not be written when FIFO\_STAT.STAT\_EMPTY is asserted. |
| STATUS | 0x0\_0054 | 0x0 | Status resister |
| STAT\_WD\_CTRL | 0x0\_0080 | 0x04\_0000 | Number of clock cycles (interface domain) before the status watchdog interrupt triggers. The reset value of this register is: 0x04\_0000 The minimum value this register may be set to is 1. |
| KEY\_SZ | 0x0\_0100 | 0x0 | Key size |
| VSPACC\_RQST | 0x0\_0140 | 0x0 | Any write to this register causes a request of the virtual SPAcc resource to be allocated. |
| VSPACC\_ALLOC | 0x0\_0144 | 0x0 | Allocation state of the virtual SPAcc resource. 0=free, 1=allocated, 2=busy |
| VSPACC\_PRIORITY | 0x0\_0148 | 0x100 | Control the priority of the vSPAcc |
| VSPACC\_RC4\_KEY\_REQ | 0x0\_0150 | 0x0 | Set this bit to request access to the RC4 key context for dual clock domain implementations. |
| VSPACC\_RC4\_KEY\_GNT | 0x0\_0154 | 0x0 | Indicates that the RC4 key context area is granted to the virtual SPAcc. The key context area must not be written unless this bit is set. (Dual clock domain implementations only) |
| VERSION | 0x0\_0180 | 0x0604\_8b4b | Flags indicating |
| VERSION\_EXT | 0x0\_0184 | 0x1572\_000c | Flags indicating |
| VERSION\_EXT\_2(SPAcc-QoS) | 0x0\_0190 | 0x240c\_0c0c | Configured depth of the CMDx(x=0,1,2) FIFO and the STAT FIFO. |
| VERSION\_EXT\_2(SPAcc) | 0x0\_0190 | 0X0 | Configured depth of the CMD0 FIFO and the STAT FIFO |
| SECURE\_CTRL | 0x0\_01C0 | 0X0 | Secure mode control register |
| SECURE\_CTX\_RELEASE | 0x0\_01C4 | 0X0 | Writing this register via a secure bus transaction will clear the secure state of the indicated context and allow access by insecure slave bus transactions. |
| CIPH\_KEY | 0x0\_4000-0x7FFF | 0X0 | This memory range is used for key context storage for the cipher algorithms. |
| HASH\_KEY | 0x0\_8000-0x0\_FFFF | 0X0 | This memory range is used for key context storage for the hash algorithms. |
| RC4\_CTX | 0x2\_0000-0x3\_FFFF | 0X0 | This memory range is used for accessing the internal RC4 key context. The format of this range depends on whether clock domain crossing logic is included or not. Note: In dual clock domain implementations with SPAcc virtualization turned on, this address space must be explicitly arbitrated for. |
| PKA core register | | | |
| Control | 0x0000 | 0x0 | initiate actions within the PKA |
| Program Entry Address | 0x0004 | 0x0 | indicate the start instruction word index address for the function to be executed. This register can only be written while the PKA is stopped. Writes at any other time will be ignored by the PKA. |
| Return Code | 0x0008 | 0x0 | monitor the operation of the PKA |
| Build Configuration | 0x000C | 0x4000\_9800 | contains many of the static build-time parameter enumerations |
| F-Stack Pointer | 0x0010 | 0x0 | indicates the depth of the function call stack at any point in time |
| Instructions Since Go | 0x0014 | 0x0 | Instructions executed since the specific function was started. |
| P-Stack Pointer | 0x0018 | 0x0 | indicates the depth of the parameter stack at any point in time |
| Configuration | 0x001C | 0x0 | set configuration within the PKA. |
| Status (IRQ) | 0x0020 | 0x0 | monitor and acknowledge the PKA interrupts. |
| Flags | 0x0024 | 0x0 | indicates the settings of the PKA’s flags at any point in time |
| Watchdog | 0x0028 | 0xffff\_ffff | used to provide a bounded limit on the number of instructions the PKA will execute after the assertion of CTRL\_GO. |
| Cycles Since Go | 0x002C | 0x0 | counts the number of clock cycles that have elapsed since the last CTRL\_GO was issued |
| Index I | 0x0030 | 0x0 | shows the value of the index registers |
| Index J | 0x0034 | 0x0 | shows the value of the index registers |
| Index K | 0x0038 | 0x0 | shows the value of the index registers |
| Index L | 0x003C | 0x0 | shows the value of the index registers |
| Interrupt Enable | 0x0040 | 0x0 | Enable interrupt |
| Jump Probability | 0x0044 | 0x0 | set the probability of executing a dummy operation |
| Probability LFSR Seed | 0x0048 | 0x1 | to initialize the probability LFSR |
| Bank Switch A | 0x0050 | 0x0 | shows the value of the bank switches registers at any point in time |
| Bank Switch B | 0x0054 | 0x0 | shows the value of the bank switches registers at any point in time |
| Bank Switch C | 0x0058 | 0x0 | shows the value of the bank switches registers at any point in time |
| Bank Switch D | 0x005C | 0x0 | shows the value of the bank switches registers at any point in time |
| TRNG core register | | | |
| TRNG\_CNTL | 0x0000 | 0x8000\_0000 | Control and status register that configures, starts, and monitors the TRNG engine |
| TRNG\_IRQ\_STAT | 0x0004 | 0x0 | Indicates that the TRNG has completed its task of generating a random number |
| TRNG\_IRQ\_EN | 0x0008 | 0x0 | Enables the interrupt pin to be asserted |
| TRNG\_CFG | 0x000C | 0xxx | Indicates presence or absence of the ring-based seeding mechanism. |
| TRNG\_DATA0 | 0x0010 | 0x1 | R = Random data bits 31:0  W = Nonce seed bits 31:0 and 159:128 depending on TRNG\_CNTL.NONCE\_SEED\_SELECT |
| TRNG\_DATA1 | 0x0014 | 0x0 | R = Random data bits 63:32  W = Nonce seed bits 63:32 and 191:160 depending  On TRNG\_CNTL. NONCE\_SEED\_SELECT. |
| TRNG\_DATA2 | 0x0018 | 0x0 | R = Random data bits 95:64  W = Nonce seed bits 95:64 and 223:192 depending On  TRNG\_CNTL. NONCE\_SEED\_SELECT. |
| TRNG\_DATA3 | 0x001C | 0x0 | R = Random data bits 127:96  W = Nonce seed bits 127:96 and 254:224 (only 31bits!) depending on  TRNG\_CNTL.NONCE\_SEED\_SELECT. |

* + 1. SPDU Registers

SPDU\_IRQ\_EN

Address: 0x0\_0000

Table 345.spud\_irq\_en register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | RW | 0x0 | GLBL | Global interrupt (O\_irq) control for interrupt sources. This does not affect the individual engine interrupt signals. |
| 30:22 | - | - | Reserved | |
| 21\* | RW | 0x0 | MPM | Enables interrupts from the corresponding processing engine. |
| 20\* | RW | 0x0 | EA |
| 19\* | RW | 0x0 | KEP |
| 18\* | RW | 0x0 | RE |
| 17\* | RW | 0x0 | PKA |
| 16\* | RW | 0x0 | RNG |
| 15:8 | - | - | Reserved | |
| 7:0\* | RW | 0x0 | SPAcc | Enables interrupts from the SPAcc component(s). Each virtual SPAcc component corresponds to one bit. |

SPDU\_IRQ\_STAT

Address: 0x0\_0004

Table 346.spdu\_irq\_stat register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 30:22 | - | - | Reserved | |
| 21\* | RO | 0x0 | MPM | Individual interrupt bits corresponding directly to the bits within the IRQ\_EN register. These are level sensitive and indicate the current state of the interrupts from the corresponding engine. To clear these bits the appropriate acknowledgment must be directed at the individual engine. The global O\_irq signal is asserted if the appropriate enable(s) are set and the corresponding status bits are also set. |
| 20\* | RO | 0x0 | EA |
| 19\* | RO | 0x0 | KEP |
| 18\* | RO | 0x0 | RE |
| 17\* | RO | 0x0 | PKA |
| 16\* | RO | 0x0 | RNG |
| 15:8 | - | - | Reserved |
| 7:0\* | RO | 0x0 | SPAcc |

DMA\_BRST\_SZ

Address: 0X0\_0010

Table 347.dma\_brst\_sz register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:6 | - | - | Reserved |
| 5:0 | RW | 0x10 | Maximum size of DMA burst over the master bus (in words). Reset value is 16. Minimum setting is 4. A value of zero in this register will set the burst size to maximum (64 words). This register replaces the SPAcc SDMA\_BRST\_SZ register, which is considered reserved in this design. |

SP\_DECODE

Address: 0X0\_0014

Table 348.sp\_decode register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16\* | RW | 0x0 | Upper bits of master address to decode into the scratchpad memory |
| 15:2 | - | - | Reserved |

SPACC\_CMD\_PRIORITY

Address: 0x0\_0018

Table 349.spacc\_cmd\_priority register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | - | - | Reserved |
| 1:0\* | RW | 0x0 | Set the SPAcc command priority level that the accelerator engines run at. 0 is highest (runs before CMD0 FIFO), 3 is lowest (runs after CMD2 FIFO). |

SP\_VERSION

Address: 0X0\_0180

Table 350.sp\_version register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | N/A | Copy of the SPAcc version registers. Corresponds to vSPAcc 0 if multiple virtual SPAcc engines are present |

SP\_VERSION\_EXT

Address: 0X0\_0184

Table 351.sp\_version\_ext register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | N/A | Copy of the SPAcc version registers. Corresponds to vSPAcc 0 if multiple virtual SPAcc engines are present |

SPDU\_VERSION

Address: 0X0\_0188

Table 352.spdu\_version register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:14 | - | - | Reserved |
| 13 | RO | 0x0 | Flag indicating that the MPM component is instantiated. |
| 12 | RO | 0x0 | Flag indicating that the ESP/AH component is instantiated. |
| 11 | RO | 0x0 | Flag indicating that the KEP component is instantiated. |
| 10 | RO | 0x0 | Flag indicating that the RE component is instantiated. |
| 9 | RO | 0x1 | Flag indicating that the PKA component is instantiated. |
| 8 | RO | 0x1 | Flag indicating that the RNG component is instantiated |
| 7:4 | RO | 0x2 | Major release version number of the SPAcc-PDU. |
| 3:0 | RO | 0x6 | Minor release version number of the SPAcc-PDU. |

RE

Address: 0X0\_8000-0X0\_BFFF

Table 353.RE register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | N/A | Record Engine component |

MPM

Address: 0X0\_C000-0X0\_FFFF

Table 354.MPM register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | N/A | Multi-Packet Manager component |

KEP

Address: 0X1\_0000-0X1\_3FFF

Table 355.KEP register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | N/A | Key Exchange Protocol component |

EA

Address: 0x1\_4000-0x1\_7FFF

Table 356.EA register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | 0x0 | ESP/AH Engine component |

RNG

Address: 0x1\_8000-0x1\_FFFF

Table 357.RNG register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | 0x0 | Random Number Generator component |

PKA

Address: 0x2\_0000-0x3\_FFFF

Table 358.PKA register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | 0x0 | Public Key Accelerator component |

SPACC

Address: 0x4\_0000-0x7\_FFFF

Table 359.SPACC register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | N/A | 0x0 | SPAcc component. For configurations with SPAcc virtualization turned on, this maps to the virtual SPAcc component at index 0 |

* + 1. SPAcc core Registers

IRQ\_EN

Address: Operational Base + offset (0x0\_0000)

Table 360.IRQ\_EN register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | RW | 0x0 | GLBL\_EN | Global enable control for interrupt sources. |
| 30:13 | - | - | Reserved | |
| 12 | RW | 0x0 | STAT\_WD\_EN | Enables STAT FIFO watchdog service interrupt. |
| 11:9 | - | - | Reserved | |
| 8\* | RW | 0x0 | RC4\_DMA\_EN | Enables interrupts on completion of an RC4 DMA operation when transferring context across clock domains. Valid on dual clock implementations only. |
| 7:5 | - | - | Reserved | |
| 4 | RW | 0x0 | STAT\_EN | Enables interrupts when the STAT FIFO count increments to the setting of the IRQ\_CTRL.STAT\_CNT value. |
| 3 | - | - | Reserved | |
| 2\* | RW | 0x0 | CMD2\_EN | Enables interrupts when the CMDx FIFO count decrements to the setting of the IRQ\_CTRL. CMDx\_CNT value. |
| 1\* | RW | 0x0 | CMD1\_EN |
| 0 | RW | 0x0 | CMD0\_EN |

IRQ\_STAT

Address: Operational Base + offset (0x0\_0004)

Table 361.IRQ\_STAT register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31:13 | - | - | Reserved | Individual interrupts. These correspond directly with the associated bit in the IRQ\_EN register. Write 1 to clear |
| 12 | RW | 0x0 | STAT\_WD |
| 11:9 | - | - | Reserved |
| 8\* | RW | 0x0 | RC4\_DMA |
| 7:5 | - | - | Reserved |
| 4 | RW | 0x0 | STAT |
| 3 | - | - | Reserved |
| 2\* | RW | 0x0 | CMD2 |
| 1\* | RW | 0x0 | CMD1 |
| 0 | RW | 0x0 | CMD0 |

IRQ\_CTRL(SPAcc-QoS)

Address: Operational Base + offset (0x0\_0008)

Table 362.IRQ\_CTRL(SPAcc-QoS) register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | - | - | Reserved | |
| 30:24\* | RW | 0x24 | STAT\_CNT | Configures the status FIFO count to cause a STAT interrupt to be generated when the count exceeds the threshold in this register. e.g. If set to 5, an interrupt will be generated when the FIFO count transitions from 4 to 5. The width of this field depends on the configured FIFO depth. Resets to the FIFO full condition. |
| 23:22 | - | - | Reserved | |
| 21:16\* | RW | 0x0 | CMD2\_CNT | Configures the individual command FIFO counts to cause a CMDx interrupt to be generated when the count decrements below the threshold in this register. e.g. If set to 5, an interrupt will be generated when the FIFO count transitions from 6 to 5. Resets to zero. The width of these fields depends on the configured FIFO depth. |
| 15:14 | - | - | Reserved |
| 13:8\* | RW | 0x0 | CMD1\_CNT |
| 7:6 | - | - | Reserved |
| 5:0\* | RW | 0x0 | CMD0\_CNT |

IRQ\_CTRL(SPAcc)

Address: Operational Base + offset (0x0\_0008)

Table 363.IRQ\_CTRL(SPAcc) register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31:25 | - | - | Reserved | |
| 24:16 | RW | 0x0 | STAT\_CNT | See above for SPAcc-QoS version of register. |
| 15:8 | - | - | Reserved | |
| 7:0\* | RW | 0x0 | CMD0\_CNT | See above for SPAcc-QoS version of register. |

FIFO\_STAT(SPAcc-QoS)

Address: Operational Base + offset (0x0\_000C)

Table 364. FIFO\_STAT(SPAcc-QoS) register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 0x1 | Flag indicating that the status FIFO is empty. This bit resets to 1. |
| 30:24 | - | - | Reserved |
| 23\* | RO | 0x0 | Flag indicating that the CMD2 FIFO is full. |
| 22:16\* | RO | 0x0 | Current number of commands in the CMD2 FIFO. |
| 15\* | RO | 0x0 | Flag indicating that the CMD1 FIFO is full |
| 14:8\* | RO | 0x0 | Current number of commands in the CMD1 FIFO. |
| 7 | RO | 0x0 | Flag indicating that the CMD0 FIFO is full. |
| 6:0\* | RO | 0x0 | Current number of commands in the CMD0 FIFO. |

FIFO\_STAT(SPAcc)

Address: Operational Base + offset (0x0\_000C)

Table 365.FIFO\_STAT(SPAcc) register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | RO | 0x1 | STAT\_EMPTY | Similar to SPAcc-QoS version of register. (See above) |
| 30:25 | - | - | Reserved | |
| 24:16\* | RO | 0x0 | STAT\_CNT | Similar to SPAcc-QoS version of register. (See above) |
| 15 | RO | 0x0 | CMD0\_FULL | Similar to SPAcc-QoS version of register. (See above) |
| 14:9 | - | - | Reserved | |
| 8:0\* | RO | 0x0 | CMD0\_CNT | Similar to SPAcc-QoS version of register. (See above) |

SDMA\_BRST\_SZ

Address: Operational Base + offset (0x0\_0010)

Table 366.SDMA\_BRST\_SZ register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:5 | - | - | Reserved |
| 4:0\* | RW | 0x0 | Maximum size of DMA burst over the master bus (in words). Reset value is 16. Minimum setting is 4.A value of zero in this register will set the burst size to maximum (32 words). Width configures to only 4 bits wide when SHA-384/512 algorithms are not enabled (hence reset value is zero in this case).  This register is shared globally between all virtual SPAcc components. |

SRC\_PTR

Address: Operational Base + offset (0x0\_0020)

Table 367.SRC\_PTR register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0\* | RW | 0x0 | Pointer to source packet data structure in system memory. |

DST\_PTR

Address: Operational Base + offset (0x0\_0024)

Table 368.DST\_PTR register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0\* | RW | 0x0 | Pointer to destination packet data structure in system memory. |

OFFSET

Address: Operational Base + offset (0x0\_0028)

Table 369.OFFSET register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16\* | RW | 0x0 | Offsets in bytes in the packet data structure to start of AAD |
| 15:0\* | RW | 0x0 |

PRE\_ADD\_LEN

Address: Operational Base + offset (0x0\_002C)

Table 370.PRE\_ADD\_LEN register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0\* | RW | 0x0 | Byte length of the prefixed AAD. |

POST\_ADD\_LEN

Address: Operational Base + offset (0x0\_0030)

Table 371.POST\_ADD\_LEN register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0\* | RW | 0x0 | Byte length of the post-fixed AAD. |

PROC\_LEN

Address: Operational Base + offset (0x0\_0034)

Table 372.PROC\_LEN register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0\* | RW | 0x0 | Total packet byte length for processing, including  PRE\_AAD\_LEN and payload. |

ICV\_LEN

Address: Operational Base + offset (0x0\_0038)

Table 373.ICV\_LEN register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:7 | - | - | Reserved |
| 6:0\* | RW | 0x0 | Byte length of the ICV. If set to zero, the default length for the selected hash algorithm will be used. |

ICV\_OFFSET

Address: Operational Base + offset (0x0\_003C)

Table 374.ICV\_OFFSET register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0\* | RW | 0x0 | Byte offset in packet data structure to start of ICV. Refers to SRC on decrypt, and DST on encrypt. This value is ignored if the CTRL.ICV\_APPEND flag is set. |

IV\_OFFSET

Address: Operational Base + offset (0x0\_0040)

Table 375.IV\_OFFSET register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31\* | RW | 0x0 | ENABLE | If set, the IV value will be obtained from the SRC data buffer, otherwise the IV value will be fetched from the key context page. |
| 30:0\* | RW | 0x0 | OFFSET | Byte offset in SRC data structure to obtain the cryptographic initialization value. Note: This register is only present if pIV\_IMPORT\_EN is turned on. |

SW\_CTRL

Address: Operational Base + offset (0x0\_0044)

Table 376.SW\_CTRL register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31:30\* | RW | 0x0 | PRIORITY | Selects the priority queue to place the command in (0-2). (SPAcc-QoS only) |
| 29:8 | - | - | Reserved | |
| 7:0 | RW | 0x0 | SW\_ID | Software tag of packet ID. Will be returned with the packet in the outbound status FIFO. This value automatically increments on every push to the command FIFO. |

AUX\_INFO

Address: Operational Base + offset (0x0\_0048)

Table 377.AUX\_INFO register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Selects the CTS mode to apply for CBC. (AES Only)  0: None  1: CBC-CS1  2: CBC-CS2  3: CBC-CS3 |
| 15:4 | - | - | Reserved |
| 3 | RW | 0x0 | Invert the result of the CRC. |
| 2 | RW | 0x0 | For CRC-32 operations, the complete 32-bit output result is bit reflected. |
| 1 | RW | 0x0 | For CRC-32 operations, the input bytes are bit reflected prior to computing the CRC. |
| 0\* | RW | 0x0 | For KASUMI-f9, this field is used to indicate the direction bit(uplink/downlink). |

CTRL

Address: Operational Base + offset (0x0\_004C)

Table 378.CTRL register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31\* | RW | 0x0 | SEC\_KEY | Use the secure port for fetching key information. |
| 30 | - | - | Reserved | |
| 29 | RW | 0x0 | KEY\_EXP | Expand the key on the current context. |
| 28 | RW | 0x0 | ICV\_APPEND | The ICV appears at the end of the packet. Overrides the ICV\_OFFSET register. |
| 27 | RW | 0x0 | ICV\_ENC | If set, the ICV is encrypted. ICV\_PT and ICV\_APPEND flags must be set in this case. The POST\_AAD\_LEN must also be zero |
| 26 | RW | 0x0 | ICV\_PT | If set, ICV is computed over the plaintext.  If clear, ICV is computed over the cipher text. |
| 25 | RW | 0x0 | AAD\_COPY | Copy the PRE\_AAD to the destination buffer |
| 24 | RW | 0x0 | ENCRYPT | If set, perform the encrypt transform. If clear, perform the decrypt transform |
| 23:16\* | RW | 0x0 | CTX\_IDX | Key context page index to use. |
| 15\* | RW | 0x1 | MSG\_END | Begin and end flags for partial packet processing. These bits both have reset values of 1. (SPAcc-QoS only) |
| 14\* | RW | 0x1 | MSG\_BEGIN |
| 13:12 | RW | 0x0 | HASH\_MODE | Hash mode to use. Only used for MD5 and SHA based hash algorithms. 0=Raw,1=SSLMAC, 2=HMAC |
| 11:8 | RW | 0x0 | CIPH\_MODE | Block cipher mode to use. Not used for stream ciphers. 0=ECB, 1=CBC, 2=CTR,3=CCM, 5=GCM, 7=OFB, 8=CFB, 9=f8,10=XTS |
| 7:4 | RW | 0x0 | HASH\_ALG | Hash algorithm to employ: 0=NULL,1=MD5,2=SHA-1, 3=SHA-224, 4=SHA-256, 5=SHA-384, 6=SHA-512, 7= AESXCBC-MAC, 8=AES-CMAC,9=KASUMI-f9, 10=SNOW 3G-UIA2,11=CRC-32-IEEE 802.3, 12=ZUC 128-EIA3, 13=SHA-512/224, 14=SHA-512/256  Note: The hash algorithm must be NULL if using AES-GCM or AES-CCM combined mode algorithms. |
| 3 | - | - | Reserved | |
| 2:0 | RW | 0x0 | CIPH\_ALG | Cipher algorithm to employ: 0=NULL,1=DES, 2=AES, 3=RC4, 4=MULTI2, 5=KASUMI, 6=SNOW 3G-UEA2,  7=ZUC 128-EE3 |

STAT\_POP

Address: Operational Base + offset (0x0\_0050)

Table 379.STAT\_POP register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | - | - | Reserved |
| 0 | WO | 0x0 | Writing any value to this register causes a result to be popped off of the status FIFO. Must not be written when FIFO\_STAT.STAT\_EMPTY is asserted. |

STATUS

Address: Operational Base + offset (0x0\_0054)

Table 380.STATUS register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 0x0 | Indicates if the command executed in secure or normal mode |
| 30:27 | - | - | Reserved |
| 26:24 | RO | 0x0 | Result of operation: 0=Ok, 1=ICV Fail, 2=Memory Error, 3=Block Error, 4=Security Error |
| 23:8 | - | - | Reserved |
| 7:0 | RO | 0x0 | Software tags of packet ID. |

STAT\_WD\_CTRL

Address: Operational Base + offset (0x0\_0080)

Table 381.STAT\_WD\_CTRL register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | - | - | Reserved |
| 23:0 | RW | 0x04\_0000 | Number of clock cycles (interface domain) before the status watchdog interrupt triggers. The reset value of this register is: 0x04\_0000 The minimum value this register may be set to is 1. |

KEY\_SZ

Address: Operational Base + offset (0x0\_0100)

Table 382.KEY\_SZ register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | WO | 0x0 | Indicates whether this key size is a cipher key (1) or a hash key (0). |
| 30:16 | - | - | Reserved |
| 15:8\* | WO | 0x0 | Context page index of key to set. |
| 7:0\* | WO | 0x0 | Set the size of the key in bytes. Writing this register sets the internal key size for the selected context |

VSPACC\_RQST

Address: Operational Base + offset (0x0\_0140)

Table 383.VSPACC\_RQST register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | - | - | Reserved |
| 0 | WO | 0x0 | Any write to this register causes a request of the virtual SPAcc resource to be allocated. |

VSPACC\_ALLOC

Address: Operational Base + offset (0x0\_0144)

Table 384.VSPACC\_ALLOC register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | - | - | Reserved |
| 1:0 | RO | 0x0 | Allocation state of the virtual SPAcc resource. 0=free,  1=allocated, 2=busy. |

VSPACC\_PRIORITY

Address: Operational Base + offset (0x0\_0148)

Table 385.VSPACC\_PRIORITY register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:12 | - | - | Reserved |
| 11:8 | RW | 0x1 | The weight to apply to this particular virtual SPAcc component in the priority scheme. The reset value of this weight is 1. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | Priority mode: 0=weighted,1=fixed. This field is shared globally between all virtual SPAcc components. |

VSPACC\_RC4\_KEY\_REQ

Address: Operational Base + offset (0x0\_0150)

Table 386.VSPACC\_RC4\_KEY\_REQ register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | - | - | Reserved |
| 0 | RW | 0x0 | Set this bit to request access to the RC4 key context for dual clock domain implementations. |

VSPACC\_RC4\_KEY\_GNT

Address: Operational Base + offset (0x0\_0154)

Table 387.VSPACC\_RC4\_KEY\_GNT register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | - | - | Reserved |
| 0 | RO | 0x0 | Indicates that the RC4 key context area is granted to the virtual SPAcc. The key context area must not be written unless this bit is set.(Dual clock domain implementations only) |

VERSION

Address: Operational Base + offset (0x0\_0180)

Table 388.VERSION register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | RO | 0x0604 | Project number encoded in BCD |
| 15 | RO | 0x1 | Flag indicating that partial packet processing is enabled. |
| 14:12 | RO | 0x0 | Virtual SPAcc index. |
| 11 | RO | 0x1 | Flag indicating that the AUX register is enabled. |
| 10:9 | RO | 0x1 | Type of SPAcc:  0 = SPAcc/SPAcc-QoS  1 = SPAcc-PDU  2 = SPAcc-HSM |
| 8 | RO | 0x1 | Flag indicating that QOS features are enabled. |
| 7:4 | RO | 0x4 | Major release version number of SPAcc. |
| 3:0 | RO | 0xb | Minor release version number of SPAcc. |

VERSION\_EXT

Address: Operational Base + offset (0x0\_0184)

Table 389.VERSION\_EXT register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:30 | - | - | Reserved |
| 29:28 | RO | 0x1 | DMA type configured:  1 = DDT  2 = Linear |
| 27 | - | - | Reserved |
| 26:24 | RO | 0x5 | Hash key context page size (log 2). |
| 23 | - | - | Reserved |
| 22:20 | RO | 0x0 | Cipher key context page size (log2). |
| 19:16 | RO | 0x0 | Number of virtual SPAcc engines configured. |
| 15:8 | RO | 0x0 | Number of RC4 contexts configured. |
| 7:0 | RO | 0xc | Number of contexts configured. |

VERSION\_EXT\_2(SPAcc-QoS)

Address: Operational Base + offset (0x0\_0190)

Table 390.VERSION\_EXT\_2(SPAcc-QoS) register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | - | - | Reserved |
| 30:24 | RO | 0x24 | Configured depth of the STAT FIFO. |
| 23 | - | - | Reserved |
| 22:16 | RO | 0xc | Configured depth of the CMD2 FIFO. |
| 15 | - | - | Reserved |
| 14:8 | RO | 0xc | Configured depth of the CMD1 FIFO. |
| 7 | - | - | Reserved |
| 6:0 | RO | 0xc | Configured depth of the CMD0 FIFO. |

VERSION\_EXT\_2(SPAcc)

Address: Operational Base + offset (0x0\_0190)

Table 391.VERSION\_EXT\_2(SPAcc) register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:25 | - | - | Reserved |
| 24:16 | RO | 0x24 | Configured depth of the STAT FIFO. |
| 15:9 | - | - | Reserved |
| 8:0 | RO | 0xc | Configured depth of the CMD0 FIFO. |

SECURE\_CTRL

Address: Operational Base + offset (0x0\_01C0)

Table 392.SECURE\_CTRL register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RW | 0x0 | If this bit is set, the SPAcc is inaccessible except by secure slave transactions. The reset value of this register is configured by pSECURE\_BUS\_LOCK\_RST. If any virtual SPAcc component asserts the lock, all global configuration registers may only be accessed via secure mode transactions. |
| 30\* | RW | 0x0 | This bit is only used in dual clock implementations with RC4. Setting this register prevents insecure processes from accessing the RC4 Context DMA engine. It should be set while loading/unloading secure context information to ensure data integrity and confidentiality. The reset value of this register is configured by pSECURE\_BUS\_LOCK\_RST. |
| 29:3 | - | - | Reserved |
| 2 | RW | 0x0 | These three bits configure the master bus to use secure accesses when the engine is executing a secure command. Each bit corresponds to the type of data the engine is accessing; source buffer, destination buffer, and DDT structure. The reset value of these is 1'b1. This register is only present if pSECURE\_BUS\_EN is 1. |
| 1 | RW | 0x0 |
| 0 | RW | 0x0 |

SECURE\_CTX\_RELEASE

Address: Operational Base + offset (0x0\_01C4)

Table 393.SECURE\_CTX\_RELEASE register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | - | - | Reserved |
| 7:0\* | WO | 0x0 | Writing this register via a secure bus transaction will clear the secure state of the indicated context and allow access by insecure slave bus transactions. |

CIPH\_KEY

Address: Operational Base + offset (0x0\_4000-0X7FFF)

Table 394.CIPH\_KEY register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | RW | N/A | This memory range is used for key context storage for the cipher algorithms |

HASH\_KEY

Address: Operational Base + offset (0x0\_8000-0x0\_FFFF)

Table 395.HASH\_KEY register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | RW | N/A | This memory range is used for key context storage for the hash algorithms. |

RC4\_CTX

Address: Operational Base + offset (0x2\_0000-0x3\_FFFF)

Table 396.RC4\_CTX register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| N/A | RW | N/A | This memory range is used for accessing the internal RC4 key context. The format of this range depends on whether clock domain crossing logic is included or not.  Note: In dual clock domain implementations with SPAcc virtualization turned on, this address space must be explicitly arbitrated for. |

* + 1. PKA core Register

Control

Address: Operational Base + offset (0x00)

Table 397.Control register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | RW | 0x0 | CTRL\_GO | 1 = Initiate program  0 = NOP |
| 30:28 | - | - | Reserved | |
| 27 | RW | 0x0 | CTRL\_STOP\_RQST | 1 = Request stop at next instruction  0 = NOP |
| 26:21 | - | - | Reserved | |
| 20:16 | RW | 0x0 | CTRL\_M521\_MODE | 9 = 521-bit mode (Build-Time  Optional Feature)  0 = Normal mode |
| 15:11 | - | - | Reserved | |
| 10:8 | RW | 0x0 | CTRL\_BASE\_RADIX | 0,1,7 = Reserved  6 = 4096-bit  5 = 2048-bit  4 = 1024-bit  3 = 512-bit  2 = 256-bit |
| 7:0 | RW | 0x0 | CTRL\_PARTIAL\_RADIX | See description in this section's text |

Program Entry Address

Address: Operational Base + offset (0x04)

Table 398.Program Entry Address register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| (X-1):0 | RW | 0x0 | Address of program entry point for function to execute |

Note: X will be equal to the ELP\_CLUE\_FW\_ADDR\_WIDTH macro definition.

Return Code

Address: Operational Base + offset (0x08)

Table 399.Return Code register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | RO | 0x0 | RC\_BUSY | Indicates that the PKA is actively executing a program. |
| 30 | RO | 0x0 | RC\_IRQ | Copy of the STAT\_IRQ bit in the STAT register. See STAT register for more details. |
| 29 | RO | 0x0 | RC\_WR\_PENDING | 1 = Register write operation pending.  0 = No register write operation pending. |
| 28 | RO | 0x0 | RC\_ZERO | Copy of the FLAG\_ZERO bit in the FLAGS register. |
| 27:24 | - | - | Reserved | |
| 23:16 | RO | 0x0 | RC\_STOP\_REASON | 255-10 = Reserved for firmware  9 = Operation size exceeds CFG  8 = Memory port collision  7 = P-Stack overflow  6 = P-Stack underflow  5 = Host request (CTRL\_STOP\_RQST)  4 = Watchdog  3 = F-Stack overflow  2 = F-Stack underflow  1 = Invalid op-code  0 = Normal stop |
| 15:0 | - | - | Reserved | |

Build Configuration Register

Address: Operational Base + offset (0x0C)

Table 400.Build Configuration Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31:30 | RO | 0x1 | BC\_FORMAT\_TYPE | Indicates format type 1 |
| 29:21 | - | - | Reserved | |
| 20:19 | RO | 0x0 | BC\_ALU\_SIZE | 3 = 256 bits  2 = 128 bits  1 = 64 bits  0 = 32 bits |
| 18:16 | RO | 0x0 | BC\_RSA\_SIZE | 5-7 = reserved  4 = 4096 bits  3 = 2048 bits  2 = 1024 bits  1 = 512 bits  0 = Not present |
| 15:14 | RO | 0x2 | BC\_ECC\_SIZE | 3 = 1024 bits  2 = 512 bits  1 = 256 bits  0 = Not present |
| 13:11 | RO | 0x3 | BC\_FW\_ROM | 5-7 = reserved  4 = 2048 words  3 = 1024 words  2 = 512 words  1 = 256 words  0 = Not present |
| 10:8 | RO | 0x0 | BC\_FW\_RAM | 5-7 = reserved  4 = 2048 words  3 = 1024 words  2 = 512 words  1 = 256 words  0 = 0 words |
| 7:0 | RO | 0x0 | BC\_BANK\_SW\_A/B/C/D | 3 = reserved  2 = 4 banks  1 = 2 banks  0 = 1 bank |

Function Stack Pointer Register

Address: Operational Base + offset (0x10)

Table 401.Function Stack Pointer Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | Current number of function stack slots in use |

Instructions Since

Address: Operational Base + offset (0x14)

Table 402.Instructions Since Go register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0x0 | Instructions executed since the specific function was started. |

Parameter Stack Pointer

Address: Operational Base + offset (0x18)

Table 403.Parameter Stack Pointer register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:6 | - | - | Reserved. |
| 5:0 | RW | 0x0 | Current number of parameter stack slots in use. |

Configuration

Address: Operational Base + offset (0x1C)

Table 404.Configuration register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:27 | - | - | Reserved |
| 26 | RW | 0x0 | 1 = Byte Lane Swap (big endian)  0 = No Byte Lane Swap (little endian) |
| 25:1 | - | - | Reserved |
| 0 | RW | 0x0 | 1 = Switch the C-memory read access slot to the DPA shadow memory.(Primarily exists for synthesis purposes.)  0 = Normal C-memory access |

Status

Address: Operational Base + offset (0x20)

Table 405.Status register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | - | - | Reserved | |
| 30 | RW | 0x0 | STAT\_IRQ | W1 = Acknowledge existing interrupt  W0 = NOP  R1 = Unacknowledged interrupt exists.  R0 = No unacknowledged interrupt exists. |
| 29:0 | - | - | Reserved | |

Flags

Address: Operational Base + offset (0x24)

Table 406.Flags register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | - | - | Reserved |
| 7 | RW | 0x0 | FLAG\_F3 |
| 6 | RW | 0x0 | FLAG\_F2 |
| 5 | RW | 0x0 | FLAG\_F1 |
| 4 | RW | 0x0 | FLAG\_F0 |
| 3 | RW | 0x0 | FLAG\_CARRY(C) |
| 2 | RW | 0x0 | FLAG\_BORROW(B) |
| 1 | RW | 0x0 | FLAG\_MEMBIT(M) |
| 0 | RW | 0x0 | FLAG\_ZERO(Z) |

Watchdog

Address: Operational Base + offset (0x28)

Table 407.Watchdog register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0xFFFF\_FFFF | INSTRUCTIONS\_UNTIL\_HALT |

Cycles Since Go

Address: Operational Base + offset (0x2C)

Table 408.Cycles Since Go register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | counts the number of clock cycles |

Index I

Address: Operational Base + offset (0x30)

Table 409.Index I register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0 | value of the index |

Index J

Address: Operational Base + offset (0x34)

Table 410.Index J register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0 | value of the index |

Index k

Address: Operational Base + offset (0x38)

Table 411.Index k register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0 | value of the index |

Index L

Address: Operational Base + offset (0x3C)

Table 412.Index L register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0 | value of the index |

Interrupt Enable

Address: Operational Base + offset (0x40)

Table 413.Interrupt Enable register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | - | - | Reserved |
| 30 | RW | 0x0 | 1 = Enable STAT\_IRQ to generate an interrupt signal at O\_irq.  0 = Disable interrupt signal at O\_irq. |
| 29:0 | - | - | Reserved |

Jump Probability

Address: Operational Base + offset (0x44)

Table 414.Jump Probability register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:13 | - | - | Reserved |
| 12:0 | RW | 0x0 | 13'h1FFF = 100% probability. 0 = 0% probability.  Other = 100% \* Other/13'h1FFF probability. |

Probability LFSR Seed

Address: Operational Base + offset (0x48)

Table 415.Probability LFSR Seed register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:13 | - | - | Reserved |
| 12:0 | RW | 0x1 | Seed to write to Probability LFSR.  Seed of 0 will automatically be  detected by the PKA H/W and written as 1 to prevent LFSR lockup |

Bank Switch A

Address: Operational Base + offset (0x50)

Table 416.Bank Switch A register

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | | **Description** | |
| 31:2 | - | | - | | Reserved |
| 1:0 | RW | 0x0 | | BANK\_SW\_A | |

Bank Switch B

Address: Operational Base + offset (0x54)

Table 417.Bank Switch B register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | - | - | Reserved |
| 1:0 | RW | 0x0 | BANK\_SW\_B |

Bank Switch C

Address: Operational Base + offset (0x58)

Table 418.Bank Switch register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | - | - | Reserved |
| 1:0 | RW | 0x0 | BANK\_SW\_C |

Bank Switch D

Address: Operational Base + offset (0x5C)

Table 419.Bank Switch register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | - | - | Reserved |
| 1:0 | RW | 0x0 | BANK\_SW\_D |

* + 1. TRNG Inside Register

Control Register

Address: Operational Base + offset (0x00)

Table 420.Control Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** | |
| 31 | RW | 0x0 | RAND\_RESEED | R1 = Random reseeding operation underway  R0 = Normal operation.  W1 = Initiate random reseeding operation  (can only be initiated if RAND\_RESEED is not high)  W0 = NOP |
| 30 | RW | 0x0 | NONCE\_RESEED | 1 = Execute nonce reseed operation.  0 = Disable nonce reseed operation. |
| 29 | WO | 0x0 | NONCE\_RESEED\_LD | Load TRNG\_DATA[0-3] registers into portion of nonce selected by the NONCE\_SEED\_SELECT bit. |
| 28 | WO | 0x0 | NONCE\_SEED\_SELECT | W1 = Select 127-bit upper portion of nonce to reseed.  W0 = Select 128-bit lower portion of nonce to reseed. |
| 27:1 | - | - | Reserved | Set to 0. |
| 0 | RW | 0x0 | GEN\_NEW\_RANDOM | R1 = Random generation underway.  R0 = No random generation underway.  W1 = Generate new random number.  W0 = NOP |

TRNG\_IRQ\_STAT

Address: Operational Base + offset (0x04)

Table 421.TRNG\_IRQ\_STAT register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | - | - | Reserved. Set to 0 |
| 27 | RW | 0x0 | R1 = TRNG interrupt/done is asserted  R0 = TRNG interrupt/done is not asserted  W1= acknowledge TRNG interrupt/done, bit is cleared |
| 26:0 | - | - | Reserved. Set to 0 |

TRNG\_IRQ\_EN

Address: Operational Base + offset (0x08)

Table 422.TRNG\_IRQ\_EN register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:28 | - | - | Reserved. Set to 0 |
| 27 | RW | 0x0 | 1 = Enable the TRNG interrupt pin  0 = Disable the TRNG interrupt pin |
| 26:0 | - | - | Reserved. Set to 0 |

TRNG\_CFG

Address: Operational Base + offset (0x0C)

Table 423.TRNG\_CFG register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:30 | RO | 0x0 | 0: Manual seeding disabled. Self seed only.  1: Self seed circuitry is present.  2: Self seed circuitry not present.  Other: Reserved |
| 29:0 | - | - | Reserved. Set to 0 |

TRNG DataA0

Address: Operational Base + offset (0x10)

Table 424.TRNG DataA0 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0x1 | R = 32-bit component of 128-bit random number  W = Word 0 of either NONCEH or NONCEL |

TRNG DataA1

Address: Operational Base + offset (0x14)

Table 425.TRNG DataA1 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0x0 | R = 32-bit component of 128-bit random number  W = Word 0 of either NONCEH or NONCEL |

TRNG DataA2

Address: Operational Base + offset ( 0x18)

Table 426.TRNG DataA2 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0x0 | R = 32-bit component of 128-bit random number  W = Word 0 of either NONCEH or NONCEL |

TRNG DataA3

Address: Operational Base + offset (0x1c)

Table 427.TRNG DataA3 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0x0 | R = 32-bit component of 128-bit random number  W = Word 0 of either NONCEH or NONCEL |

* 1. Programming Sequence

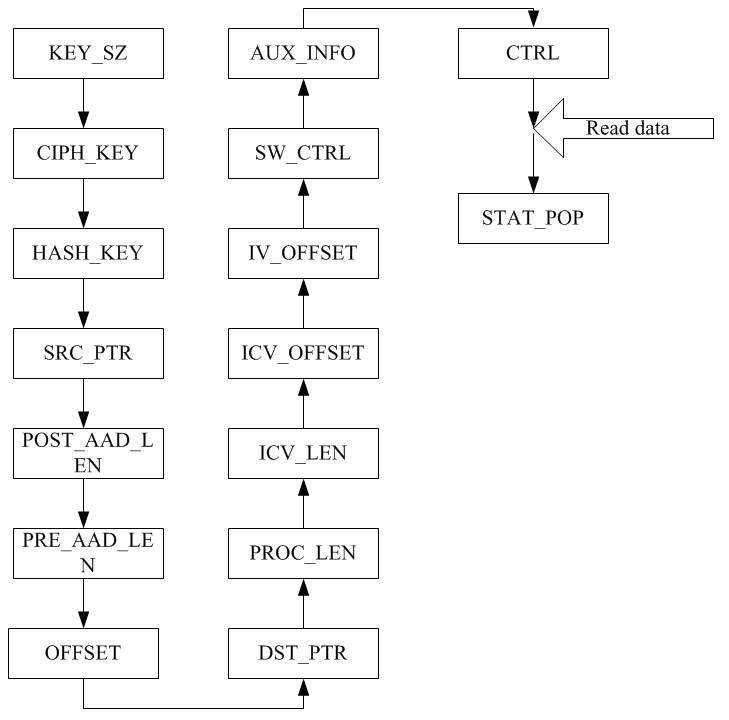


Figure 10.Crypto Engine Programming Sequence

1. Choose key mode (Hash/cipher) configure KEY\_SZ register.
2. Put the cipher key and hash key to memory, configure the CIPH\_KEY register.
3. Source pointer using SRC\_PTR register.
4. Setup post/pre data length using POST\_AAD\_LEN/PRE\_AAD\_LEN register.
5. Offset in packet data structure to start of ADD.
6. Setup payload length using PROC\_LEN register.
7. Setup integrity checking length using ICV\_LEN register.
8. Setup Initial value offset, using IV\_OFFSET register.
9. Set priority using SW\_CTRL register.
10. Setup CBC mode using AUX\_INFO register.
11. Setup CTRL register (CIPH\_ALG, CIPH\_MODE,ICV\_ENC,ENCRYPT).
12. Read data from registers.
13. Setup STAT\_POP register.
14. DMA Engine
    1. Register Summary

Table 428.DMA Engine Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| DSR | 0x000 | 0x00000000 | Returns information about the status of the DMA manager thread |
| DPC | 0x004 | 0x00000000 | Provides the value of the program counter for the DMA manager thread |
| INTEN | 0x020 | 0x00000000 | When the DMAC executes a DMASEV instruction, each bit of the INTEN Register controls if the DMAC signals |
| INT\_EVENT\_RIS | 0x024 | 0x00000000 | Returns the status of the event-interrupt resources |
| INTMIS | 0x028 | 0x00000000 | Provides the status of the active interrupts in the DMAC |
| INTCLR | 0x02c | 0x00000000 | Provides the status of the active interrupts in the DMAC |
| FSRD | 0x030 | 0x00000000 | Provides the fault status of the DMA manager |
| FSRC | 0x034 | 0x00000000 | Provides the fault status for the DMA channels |
| FTRD | 0x038 | 0x00000000 | Provides the type of fault that occurred to move the DMA manager to the Faulting state |
| FTR0 | 0x040 | 0x00000000 | Fault type for DMA channel 0 |
| FTR1 | 0x044 | 0x00000000 | Fault type for DMA channel 1 |
| FTR2 | 0x048 | 0x00000000 | Fault type for DMA channel 2 |
| FTR3 | 0x04c | 0x00000000 | Fault type for DMA channel 3 |
| FTR4 | 0x050 | 0x00000000 | Fault type for DMA channel 4 |
| FTR5 | 0x054 | 0x00000000 | Fault type for DMA channel 5 |
| FTR6 | 0x058 | 0x00000000 | Fault type for DMA channel 6 |
| FTR7 | 0x05c | 0x00000000 | Fault type for DMA channel 7 |
| CSR0 | 0x100 | 0x00000000 | Channel status for DMA channel 0 |
| CSR1 | 0x108 | 0x00000000 | Channel status for DMA channel 1 |
| CSR2 | 0x110 | 0x00000000 | Channel status for DMA channel 2 |
| CSR3 | 0x118 | 0x00000000 | Channel status for DMA channel 3 |
| CSR4 | 0x120 | 0x00000000 | Channel status for DMA channel 4 |
| CSR5 | 0x128 | 0x00000000 | Channel status for DMA channel 5 |
| CSR6 | 0x130 | 0x00000000 | Channel status for DMA channel 6 |
| CSR7 | 0x138 | 0x00000000 | Channel status for DMA channel 7 |
| CPC0 | 0x104 | 0x00000000 | Channel PC for DMA channel 0 |
| CPC1 | 0x10c | 0x00000000 | Channel PC for DMA channel 1 |
| CPC2 | 0x114 | 0x00000000 | Channel PC for DMA channel 2 |
| CPC3 | 0x11c | 0x00000000 | Channel PC for DMA channel 3 |
| CPC4 | 0x124 | 0x00000000 | Channel PC for DMA channel 4 |
| CPC5 | 0x12c | 0x00000000 | Channel PC for DMA channel 5 |
| CPC6 | 0x134 | 0x00000000 | Channel PC for DMA channel 6 |
| CPC7 | 0x13c | 0x00000000 | Channel PC for DMA channel 7 |
| SAR0 | 0x400 | 0x00000000 | Source address for DMA channel 0 |
| SAR1 | 0x420 | 0x00000000 | Source address for DMA channel 1 |
| SAR2 | 0x440 | 0x00000000 | Source address for DMA channel 2 |
| SAR3 | 0x460 | 0x00000000 | Source address for DMA channel 3 |
| SAR4 | 0x480 | 0x00000000 | Source address for DMA channel 4 |
| SAR5 | 0x4a0 | 0x00000000 | Source address for DMA channel 5 |
| SAR6 | 0x4c0 | 0x00000000 | Source address for DMA channel 6 |
| SAR7 | 0x4e0 | 0x00000000 | Source address for DMA channel 7 |
| DAR0 | 0x404 | 0x00000000 | Destination address for DMA channel 0 |
| DAR1 | 0x424 | 0x00000000 | Destination address for DMA channel 1 |
| DAR2 | 0x444 | 0x00000000 | Destination address for DMA channel 2 |
| DAR3 | 0x464 | 0x00000000 | Destination address for DMA channel 3 |
| DAR4 | 0x484 | 0x00000000 | Destination address for DMA channel 4 |
| DAR5 | 0x4a4 | 0x00000000 | Destination address for DMA channel 5 |
| DAR6 | 0x4c4 | 0x00000000 | Destination address for DMA channel 6 |
| DAR7 | 0x4e4 | 0x00000000 | Destination address for DMA channel 7 |
| CCR0 | 0x408 | 0x00800200 | Channel control for DMA channel 0 |
| CCR1 | 0x428 | 0x00800200 | Channel control for DMA channel 1 |
| CCR2 | 0x448 | 0x00800200 | Channel control for DMA channel 2 |
| CCR3 | 0x468 | 0x00800200 | Channel control for DMA channel 3 |
| CCR4 | 0x488 | 0x00800200 | Channel control for DMA channel 4 |
| CCR5 | 0x4a8 | 0x00800200 | Channel control for DMA channel 5 |
| CCR6 | 0x4c8 | 0x00800200 | Channel control for DMA channel 6 |
| CCR7 | 0x4e8 | 0x00800200 | Channel control for DMA channel 7 |
| LC0\_0 | 0x40c | 0x00000000 | Loop counter 0 for DMA channel 0 |
| LC0\_1 | 0x42c | 0x00000000 | Loop counter 0 for DMA channel 1 |
| LC0\_2 | 0x44c | 0x00000000 | Loop counter 0 for DMA channel 2 |
| LC0\_3 | 0x46c | 0x00000000 | Loop counter 0 for DMA channel 3 |
| LC0\_4 | 0x48c | 0x00000000 | Loop counter 0 for DMA channel 4 |
| LC0\_5 | 0x4ac | 0x00000000 | Loop counter 0 for DMA channel 5 |
| LC0\_6 | 0x4cc | 0x00000000 | Loop counter 0 for DMA channel 6 |
| LC0\_7 | 0x4ec | 0x00000000 | Loop counter 0 for DMA channel 7 |
| LC1\_0 | 0x410 | 0x00000000 | Loop counter 1 for DMA channel 0 |
| LC1\_1 | 0x430 | 0x00000000 | Loop counter 1 for DMA channel 1 |
| LC1\_2 | 0x450 | 0x00000000 | Loop counter 1 for DMA channel 2 |
| LC1\_3 | 0x470 | 0x00000000 | Loop counter 1 for DMA channel 3 |
| LC1\_4 | 0x490 | 0x00000000 | Loop counter 1 for DMA channel 4 |
| LC1\_5 | 0x4b0 | 0x00000000 | Loop counter 1 for DMA channel 5 |
| LC1\_6 | 0x4d0 | 0x00000000 | Loop counter 1 for DMA channel 6 |
| LC1\_7 | 0x4f0 | 0x00000000 | Loop counter 1 for DMA channel 7 |
| DBGSTATUS | 0xd00 | 0x00000000 | Provides the debug status of the DMAC |
| DBGCMD | 0xd04 | - | Controls the execution of debug commands in the DMAC as issuing instructions to the DMAC using an APB interface |
| DBGINST0 | 0xd08 | - | Controls the debug instruction, channel, and thread information for the DMAC |
| DBGINST1 | 0xd0c | - | Controls the upper bytes of the debug instruction for the DMAC |
| CR0 | 0xe00 | 0x003FF075 | Provides the status of the tie-off control signals |
| CR1 | 0xe04 | 0x35 | Provides information about the instruction cache configuration |
| CR2 | 0xe08 | 0x0 | Provides the value of the boot address that boot\_addr[31:0] configures |
| CR3 | 0xe0c | 0xFFFFFFFF | Provides the security state of the event-interrupt resources that are initialized when the DMAC exits from reset |
| CR4 | 0xe10 | 0xFFFFFFFF | Provides the security state of the peripheral request |
| CRD | 0xe14 | 0x01F73732 | Provides information about the configuration of the data buffer, data width, and read and write issuing capability of the DMAC |
| WD | 0xe80 | 0x000000000 | Controls the watchdog behavior |
| periph\_id\_n | 0xfe0-0xfec | 0x00341330 | Provides information about the configuration and version of the peripheral |
| pcell\_id\_n | 0xff0-0xffc | 0xb105f00d | When concatenated, these four registers return 0xb105f00d |

* 1. Register Descriptions

DMA manger status register (DSR)

Address: DSR + offset (0x000)

Table 429.DMA manger status register (DSR)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:10 | RO | 0x0 | Read undefined. |
| 9 | RO | 0x0 | DNS  Provides the security status of the DMA manager thread:  0 = DMA manager operates in the Secure state  1 = DMA manager operates in the Non-secure state  You must use the boot\_manger\_ns signal to set the secure state of the DMA manager thread. |
| 8:4 | RO | 0x0 | Wakeup\_event  When the DMA manger thread executes a DMAWFE instruction, it was for the following event to occur:  0b00000 = event[0]  0b00001 = event[1]  0b00010 = event[2] |
| 3:0 | RO | 0x0 | DMA status  The operating state of the DMA manager:  0b0000 = Stopped  0b0001 = Executing  0b0010 = Cache miss  0b0011 = Updating PC  0b0100 = Waiting for event  0b0101-0b1110 = reserved  0b1111 = Faulting. |

DMA program counter register (DPC)

Address: DPC + offset (0x004)

Table 430.DMA program counter register (DPC)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | Program counter for the DMA manger thread. |

Interrupt enable register (INTEN)

Address: INTEN + offset (0x020)

Table 431.Interrupt enable register (INTEN)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0x0 | event\_irq\_select  Program the appropriate bit to control how the DMAC responds when it executes DMASEV:  Bit[N] = 0  If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit[N] to 0 if your system design does not use irq[N] to signal an interrupt request.  Bit[N] = 1  If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set is [N] to 1 if your system design requires irq[N] to signal an interrupt request. |

Event-Interrupt raw status register (INT\_EVENT\_RIS)

Address: INT\_EVENT\_RIS + offset (0x024)

Table 432.Event-Interrupt raw status register (INT\_EVENT\_RIS)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | DMASEV active  Returns the status of the event-interrupt resources:  Bit[N] = 0 Event N is inactive or irq[N] is LOW  Bit[N] = 1 Event N is active or irq[N] is HIGH  Note:  When the DMAC executes a DMASEV N instruction to send event N, the INTEN register controls whether the DMAC:  • Signals an interrupt using the appropriate irq.  • Sends the event to all of the threads. |

Interrupt status register (INTMIS)

Address: INTMIS + offset (0x028)

Table 433.Interrupt status register (INTMIS)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | Irq\_status  Provides the status of the interrupts that are active in the DMAC:  Bit[N] = 0 Interrupt N is inactive and therefore irq[N] is LOW.  Bit[N] = 1 Interrupt N is active and therefore irq[N] is HIGH. |

Interrupt clear register (INTCLR)

Address: INTCLR + offset (0x02C)

Table 434.Interrupt clear register (INTCLR)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | WO | 0x0 | irq\_clr  Controls the clearing of the irq outputs:  Bit[N] = 0 The status of irq[N] does not change.  Bit[N] = 1 The DMAC sets irq[N] LOW if the INTEN register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change. |

Fault status DMA manger register (FSRD)

Address: FSRD + offset (0x030)

Table 435.Fault status DMA manger register (FSRD)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | RO | 0x0 | Reserved, read undefined. |
| 0 | RO | 0x0 | fs\_mgr  Provides the fault status of the DMA manger. Read as:  0 = the DMA manager thread is not in the Faulting state  1 = the DMA manager thread is in the Faulting state. |

Fault status DMA channel register (FSRC)

Address: FSRC + offset (0x034)

Table 436.Fault status DMA channel register (FSRC)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined |
| 7:0 | RO | 0x0 | fault\_status  Each bit provides the fault status of the corresponding channel. Read as :  Bit[N] = 0 No fault is present on DMA channel N.  Bit[N] = 1 DMA channel N is in the Faulting or Faulting completing state. |

Fault type DMA manager register (FTRD)

Address: FTRD + offset (0x038)

Table 437.Fault type DMA manager register (FTRD)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 0x0 | Read undefined |
| 30 | RO | 0x0 | dbg\_instr  If the DMA manager aborts, this bit indicates whether the erroneous instruction was read from the system memory or from the debug interface:  0 = instruction that generate an abort was read from system memory  1 = instruction that generate an abort was read from the debug interface. |
| 29:17 | RO | 0x0 | Read undefined. |
| 16 | RO | 0x0 | instr\_fetch\_err  Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA manager performs an instruction fetch:  0 = OKAY response  1 = EXOKAY, SLVERR, or DECERR response. |
| 15:6 | RO | 0x0 | Read undefined. |
| 5 | RO | 0x0 | mgr\_evnt\_err  Indicates whether the DMA manager was attempting to execute DMAWFE or DMASEV with inappropriate security permissions:  0 = the DMA manager has appropriate security to execute DMAWFE or DMASEV  1 = a DMA manager thread in the Non-secure state attempted to execute either:  • DMAWFE to wait for a secure event.  • DMASEV to create a secure event or secure interrupt. |
| 4 | RO | 0x0 | dmago\_err  Indicates whether the DMA manager was attempting to execute DMAGO with inappropriate security permissions:  0 = the DMA manager has appropriate security to execute DMAGO  1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state. |
| 3:2 | RO | 0x0 | Read undefined. |
| 1 | RO | 0x0 | operand\_invalid  Indicates whether the DMA manager was attempting to execute an instruction operand that was not valid for the configuration of the DMAC:  0 = valid operand  1 = invalid operand. |
| 0 | RO | 0x0 | undef\_instr  Indicates whether the DMA manager was attempting to execute an undefined instruction:  0 = defined instruction  1 = undefined instruction. |

Fault type DMA channel register (FTRn)

Address: FTRn + offset (0x040~05C)

Table 438.Fault type DMA channel register (FTRn)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 0x0 | lockup\_err  Indicates whether the DMA channel has locked-up because of resource starvation:  0 = DMA channel has adequate resources  1 = DMA channel has locked-up because of insufficient resources.  This fault is an imprecise abort. |
| 30 | RO | 0x0 | dbg\_instr  If the DMA channel aborts, this bit indicates whether the erroneous instruction was read from the  system memory or from the debug interface:  0 = instruction that generated an abort was read from system memory  1 = instruction that generated an abort was read from the debug interface.  This fault is an imprecise abort but the bit is only valid when a precise abort occurs. |
| 29:19 | RO | 0x0 | Reserved, read undefined. |
| 18 | RO | 0x0 | data\_read\_err  Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel  thread performs a data read:  0 = OKAY response  1 = EXOKAY, SLVERR, or DECERR response.  This fault is an imprecise abort. |
| 17 | RO | 0x0 | data\_write\_err  Indicates the AXI response that the DMAC receives on the BRESP bus, after the DMA channel  thread performs a data write:  0 = OKAY response  1 = EXOKAY, SLVERR, or DECERR response.  This fault is an imprecise abort. |
| 16 | RO | 0x0 | instr\_fetch\_err  Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel  thread performs an instruction fetch:  0 = OKAY response  1 = EXOKAY, SLVERR, or DECERR response.  This fault is a precise abort. |
| 15:14 | RO | 0x0 | Reserved, read undefined. |
| 13 | RO | 0x0 | st\_data\_unavailable  Indicates whether the MFIFO did not contain the data to enable the DMAC to perform the DMAST:  0 = MFIFO contains all the data to enable the DMAST to complete  1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete.  This fault is a precise abort. |
| 12 | RO | 0x0 | mfifo\_err  Indicates whether the MFIFO prevented the DMA channel thread from executing DMALD or DMAST.  Depending on the instruction:  DMALD  0 = MFIFO contains sufficient space  1 = MFIFO is too small to hold the data that DMALD requires.  DMAST  0 = MFIFO contains sufficient data  1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort. |
| 11:8 | RO | 0x0 | Reserved, read undefined. |
| 7 | RO | 0x0 | ch\_rdwr\_err  Indicates whether a DMA channel thread, in the Non-secure state, attempts to program the CCRn  Register to perform a secure read or secure write:  0 = a DMA channel thread in the Non-secure state is not violating the security permissions  1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write.  This fault is a precise abort. |
| 6 | RO | 0x0 | ch\_periph\_err  DMASTP, or DMAFLUSHP with inappropriate security permissions:  0 = a DMA channel thread in the Non-secure state is not violating the security permissions  1 = a DMA channel thread in the Non-secure state attempted to execute either:  • DMAWFP to wait for a secure peripheral.  • DMALDP or DMASTP to notify a secure peripheral.  • DMAFLUSHP to flush a secure peripheral.  This fault is a precise abort. |
| 5 | RO | 0x0 | ch\_evnt\_err  Indicates whether the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate  security permissions:  0 = a DMA channel thread in the Non-secure state is not violating the security permissions  1 = a DMA channel thread in the Non-secure state attempted to execute either:  • DMAWFE to wait for a secure event.  • DMASEV to create a secure event or secure interrupt.  This fault is a precise abort. |
| 4:2 | RO | 0x0 | Reserved, read undefined. |
| 1 | RO | 0x0 | operand\_invalid  Indicates whether the DMA channel thread was attempting to execute an instruction operand that was  not valid for the configuration of the DMAC:  0 = valid operand  1 = invalid operand.  This fault is a precise abort. |
| 0 | RO | 0x0 | undef\_instr  Indicates whether the DMA channel thread was attempting to execute an undefined instruction:  0 = defined instruction  1 = undefined instruction.  This fault is a precise abort. |

Channel status registers (CSRn)

Address: CSRn + offset (0x100~138)

Table 439.Channel status registers (CSRn)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:22 | RO | 0x0 | The channel non-secure bit provides the security of the DMA channel:  0 = DMA channel operates in the Secure state  1 = DMA channel operates in the Non-secure state. |
| 21 | RO | 0x0 | CNS  The channel non-secure bit provides the security of the DMA channel:  0 = DMA channel operates in the Secure state  1 = DMA channel operates in the Non-secure state. |
| 20:16 | RO | 0x0 | Reserved, read undefined. |
| 15 | RO | 0x0 | dmawfp\_periph  When the DMA channel thread executes DMAWFP, this bit indicates whether the periph operand was set:  0 = DMAWFP executed with the periph operand not set  1 = DMAWFP executed with the periph operand set. |
| 14 | RO | 0x0 | dmawfp\_b\_ns  When the DMA channel thread executes DMAWFP, this bit indicates whether the burst or single operand  were set:  0 = DMAWFP executed with the single operand set  1 = DMAWFP executed with the burst operand set. |
| 13:9 | RO | 0x0 | Reserved, read undefined. |
| 8:4 | RO | 0x0 | Wakeup number  If the DMA channel is in the Waiting for event state, or the Waiting for peripheral state, then these bits  indicate the event or peripheral number that the channel is waiting for:  0b00000 = DMA channel is waiting for event, or peripheral, 0  0b00001 = DMA channel is waiting for event, or peripheral, 1  0b00010 = DMA channel is waiting for event, or peripheral, 2  ……  0b11111 = DMA channel is waiting for event, or peripheral, 31. |
| 3:0 | RO | 0x0 | Channel status  The channel status encoding is:  0b0000 = Stopped  0b0001 = Executing  0b0010 = Cache miss  0b0011 = Updating PC  0b0100 = Waiting for event  0b0101 = At barrier  0b0110 = reserved  0b0111 = Waiting for peripheral  0b1000 = Killing  0b1001 = Completing  0b1010-0b1101 = reserved  0b1110 = Faulting completing  0b1111 = Faulting. |

Channel program counter register (CPCn)

Address:

CPCn + offset (0x104~13C)

Table 440.Channel program counter register (CPCn)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | pc\_chnl  Program counter for the DMA channel n thread, where n depends on the address of the register. |

Source address register (SARn)

Address: SARn + offset (0x400~4E0)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | src\_addr  Address of the source data for DMA channel n, where n depends on the address of the register. |

Destination address registers (DARn)

Address: DARn + offset (0x404~4E4)

Table 441.Destination address registers(DARn)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RO | 0x0 | dst\_addr  Address for the destination data for DMA channel n, where n depends on the address of the register. |

Channel control register (CCRn)

Address: CCRn + offset (0x408~4E8)

Table 442.Channel control register(CCRn)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RO | 0x0 | Reserved, read undefined. |
| 30:28 | RO | 0x0 | endian\_swap\_size  Endian swap size  0b000 No swap, 8-bit data  0b001 Swap bytes within 16-bit data  0b010 Swap bytes within 32-bit data  0b011 Swap bytes within 64-bit data  0b100 Swap bytes within 128-bit data  0b101 Reserved  0b110 Reserved  0b111 Reserved. |
| 27:25 | RO | 0x0 | dst\_cache\_ctrl  Programs the state of AWCACHE[3,1:0]a when the DMAC writes the destination data.  Bit [27] 0 = AWCACHE[3] is LOW  1 = AWCACHE[3] is HIGH.  Bit [26] 0 = AWCACHE[1] is LOW  1 = AWCACHE[1] is HIGH.  Bit [25] 0 = AWCACHE[0] is LOW  1 = AWCACHE[0] is HIGH. |
| 24:22 | RO | 0x0 | dst\_prot\_ctrl  Programs the state of AWPROT[2:0]a when the DMAC writes the destination data.  Bit [24] 0 = AWPROT[2] is LOW  1 = AWPROT[2] is HIGH.  Bit [23] 0 = AWPROT[1] is LOW  1 = AWPROT[1] is HIGH.  Bit [22] 0 = AWPROT[0] is LOW  1 = AWPROT[0] is HIGH. |
| 21:18 | RO | 0x0 | dst\_burst\_len  For each burst, these bits program the number of data transfers that the DMAC performs when it writes  the destination data:  0b0000 = 1 data transfer  0b0001 = 2 data transfers  0b0010 = 3 data transfers  ……  0b1111 = 16 data transfers.  The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction  is the product of dst\_burst\_len and dst\_burst\_size. |
| 17:15 | RO | 0x0 | dst\_burst\_size  For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination:  0b000 = writes 1 byte per beat  0b001 = writes 2 bytes per beat  0b010 = writes 4 bytes per beat  0b011 = writes 8 bytes per beat  0b100 = writes 16 bytes per beat  0b101-0b111 = reserved.  The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction  is the product of dst\_burst\_len and dst\_burst\_size. |
| 14 | RO | 0x0 | dst\_inc  Programs the burst type that the DMAC performs when it writes the destination data:  0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW.  1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH. |
| 13:11 | RO | 0x0 | src\_cache\_ctrl  Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data.  Bit [13] 0 = ARCACHE[2] is LOW  1 = ARCACHE[2] is HIGH.  Bit [12] 0 = ARCACHE[1] is LOW  1 = ARCACHE[1] is HIGH.  Bit [11] 0 = ARCACHE[0] is LOW  1 = ARCACHE[0] is HIGH. |
| 10:8 | RO | 0x0 | scr\_prot\_ctrl  Programs the state of ARPROT[2:0]a when the DMAC reads the source data.  Bit [10] 0 = ARPROT[2] is LOW  1 = ARPROT[2] is HIGH.  Bit [9] 0 = ARPROT[1] is LOW  1 = ARPROT[1] is HIGH.  Bit [8] 0 = ARPROT[0] is LOW  1 = ARPROT[0] is HIGH. |
| 7:4 | RO | 0x0 | src\_burst\_len  For each burst, these bits program the number of data transfers that the DMAC performs when it reads  the source data:  0b0000 = 1 data transfer  0b0001 = 2 data transfers  0b0010 = 3 data transfers  ……  0b1111 = 16 data transfers.  The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction  is the product of src\_burst\_len and src\_burst\_size. |
| 3:1 | RO | 0x0 | src\_burst\_size  For each beat within a burst, it programs the number of bytes that the DMAC reads from the source:  0b000 = reads 1 byte per beat  0b001 = reads 2 bytes per beat  0b010 = reads 4 bytes per beat  0b011 = reads 8 bytes per beat  0b100 = reads 16 bytes per beat  0b101-0b111 = reserved.  The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src\_burst\_len and src\_burst\_size. |
| 0 | RO | 0x0 | src\_inc  Programs the burst type that the DMAC performs when it reads the source data:  0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW.  1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH. |

Loop counter 0 registers (LC0\_n)

Address: LC0\_n + offset (0x40C~4EC)

Table 443.Loop counter 0 registers (LC0\_n)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:0 | RO | 0x0 | Loop counter iterations  The number of loop counter iterations. |

Loop counter 1 registers (LC1\_n)

Address: LC1\_n + offset (0x410~4F0)

Table 444.Loop counter 1 registers (LC1\_n)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:0 | RO | 0x0 | Loop counter iterations  The number of loop counter iterations. |

Debug status register (DBGSTATUS)

Address: DBGSTATUS + offset (0xd00)

Table 445.Debug status register(DBGSTATUS)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:1 | RO | 0x0 | Reserved, read undefined. |
| 0 | RO | 0x0 | dbgstatus  The debug status encoding is:  0 = Idle  1 = Busy. |

Debug command register (DBGCMD)

Address: DBGCMD + offset (0xd04)

Table 446.Debug command register (DBGCMD)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:2 | WO | - | Reserved, write as zero |
| 1:0 | WO | - | dbgcmd  The debug encoding is as follows:  0b00 = execute the instruction that the DBGINST [1:0] Registers contain  0b01 = reserved  0b10 = reserved  0b11 = reserved. |

Debug instruction-0 register (DBGINST0)

Address: DBGINST0 + offset (0xd08)

Table 447.Debug instruction-0 register(DBGINST0)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | WO | - | Instruction byte 1. |
| 23:16 | WO | - | Instruction byte 0. |
| 15:11 | WO | - | Reserved. Write as zero. |
| 10:8 | WO | - | Channel number  DMA channel number:  0b000 = DMA channel 0  0b001 = DMA channel 1  0b010 = DMA channel 2  ……  0b111 = DMA channel 7. |
| 7:1 | WO | - | Reserved. Write as zero. |
| 0 | WO | - | debug thread  The debug thread encoding is as follows:  0 = DMA manger thread  1 = DMA channel. |

Debug instruction-1 register (DBGINST1)

Address: DBGINST1 + offset (0xd0c)

Table 448.Debug instruction-1 register(DBGINST1)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | WO | - | Instruction byte 5. |
| 23:16 | WO | - | Instruction byte 4. |
| 15:8 | WO | - | Instruction byte 3. |
| 7:0 | WO | - | Instruction byte 2. |

Configuration register 0 (CR0)

Address: CR0 + offset (0xe00)

Table 449.Configuration register 0 (CR0)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:22 | RO | 0x0 | Reserved, read undefined |
| 21:17 | RO | 0x1F | num\_events  Number of interrupt outputs that the DMAC provides:  0b00000 = 1 interrupt output, irq[0]  0b00001 = 2 interrupt outputs, irq[1:0]  0b00010 = 3 interrupt outputs, irq[2:0]  ……  0b11111 = 32 interrupt outputs, irq[31:0]. |
| 16:12 | RO | 0x1F | num\_periph\_req  Number of peripheral request interfaces that the DMAC provides:  0b00000 = 1 peripheral request interface  0b00001 = 2 peripheral request interfaces  0b00010 = 3 peripheral request interfaces  ……  0b11111 = 32 peripheral request interfaces. |
| 11:7 | RO | 0x0 | Reserved, read undefined. |
| 6:4 | RO | 0x7 | num\_chnls  Number of DMA channels that the DMAC supports:  0b000 = 1 DMA channel  0b001 = 2 DMA channels  0b010 = 3 DMA channels  ……  0b111 = 8 DMA channels. |
| 3 | RO | 0x0 | Reserved, read undefined. |
| 2 | RO | 0x1 | mgr\_ns\_at\_rst  Indicates the status of the boot\_manager\_ns signal when the DMAC exited from reset:  0 = boot\_manager\_ns was LOW  1 = boot\_manager\_ns was HIGH. |
| 1 | RO | 0x0 | boot\_en  Indicates the status of the boot\_from\_pc signal when the DMAC exited from reset:  0 = boot\_from\_pc was LOW  1 = boot\_from\_pc was HIGH. |
| 0 | RO | 0x1 | preiph\_req  Supports peripheral requests:  0 = the DMAC does not provide a peripheral request interface.  1 = the DMAC provides the number of peripheral request interfaces that the num\_periph\_req field specifies. |

Configuration register 1 (CR1)

Address: CR1 + offset (0xe04)

Table 450.Configuration register 1 (CR1)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:4 | RO | 0x3 | num\_i-cache\_lines  Number of i-cache lines:  0b0000 = 1 i-cache line  0b0001 = 2 i-cache lines  0b0010 = 3 i-cache lines.  ……  0b1111 = 16 i-cache lines. |
| 3 | RO | 0x0 | Reserved, read undefined. |
| 2:0 | RO | 0x5 | i-cache\_len  The length of an i-cache line:  0b000-0b001 = reserved  0b010 = 4 bytes  0b011 = 8 bytes  0b100 = 16 bytes  0b101 = 32 bytes  0b110-0b111 = reserved. |

Configuration register 2 (CR2)

Address: CR2 + offset (0xe08)

Table 451.Configuration register 2 (CR2)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RO | 0x0 | boot\_addr  Provides the value of boot\_addr[31:0] when the DMAC exited from reset. |

Configuration register 3 (CR3)

Address: CR3 + offset (0xe0c)

Table 452.Configuration register 3 (CR3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RO | 0xFFFFFFFF | INS  Provides the security state of an event-interrupt resource:  Bit [N] = 0 Event[N] or irq[N] is in the Secure state.  Bit [N] = 1 Event[N] or irq[N] is in the Non-secure state. |

Configuration register 4 (CR4)

Address: CR4 + offset (0xe10)

Table 453.Configuration register 4 (CR4)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RO | 0xFFFFFFFF | PNS  Provides the security state of the peripheral request interfaces:  Bit [N] = 0 Peripheral request interface N is in the Secure state.  Bit [N] = 1 Peripheral request interface N is in the Non-secure state. |

DMA configuration register (CRD)

Address: CRD + offset (0xe14)

Table 454.DMA configuration register (CRD)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:30 | RO | 0x0 | Reserved, read undefined. |
| 29:20 | RO | 0x01F | data\_buffer\_dep  The number of lines that the data buffer contains:  0b000000000 = 1 line  0b000000001 = 2 lines  ……  0b111111111 = 1024 lines. |
| 19:16 | RO | 0x7 | rd\_q\_dep  The depth of the read queue:  0b0000 = 1 line  0b0001 = 2 lines  ……  0b1111 = 16 lines. |
| 15 | RO | 0x0 | Reserved, read undefined. |
| 14:12 | RO | 0x3 | rd\_cap  Read issuing capability that programs the number of outstanding read transactions:  0b000 = 1  0b001 = 2  ……  0b111 = 8. |
| 11:8 | RO | 0x7 | wr\_q\_dep  The depth of the write queue:  0b0000 = 1 line  0b0001 = 2 lines  ……  0b1111 = 16 lines. |
| 7 | RO | 0x0 | Reserved, read undefined. |
| 6:4 | RO | 0x3 | wr\_cap  Write issuing capability that programs the number of outstanding write transactions:  0b000 = 1  0b001 = 2  ……  0b111 = 8. |
| 3 | RO | 0x0 | Reserved, read undefined. |
| 2:0 | RO | 0x2 | data\_width  The data bus width of the AXI master interface:  0b000 = reserved  0b001 = reserved  0b010 = 32-bit  0b011 = 64-bit  0b100 = 128-bit  0b101-0b111 = reserved. |

Watchdog register (WD)

Address: WD + offset (0xe80)

Table 455.Watchdog register(WD)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:1 | RW | 0x0 | Reserved, read undefined. |
| 0 | RW | 0x0 | wd\_irq\_only  Controls how the DMAC responds when it detects a lock-up condition:  0 = the DMAC aborts all of the contributing DMA channels and sets irq\_abort HIGH  1 = the DMAC sets irq\_abort HIGH. |

Peripheral identification registers (periph\_id\_n)

Address: periph\_id\_0+ offset (0xfe0)

Table 456.Peripheral identification registers (periph\_id\_0)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:0 | RO | 0x30 | part\_number\_0  Returns 0x30 |

Peripheral identification registers (periph\_id\_1)

Address: periph\_id\_1 + offset (0xfe4)

Table 457.Peripheral identification registers (periph\_id\_1)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:4 | RO | 0x1 | designer\_0 Return 0x1 |
| 3:0 | RO | 0x3 | part\_number\_1 Returns 0x3 |

Peripheral identification registers (periph\_id\_2)

Address: periph\_id\_2 + offset (0xfe8)

Table 458.Peripheral identification registers (periph\_id\_2)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:4 | RO | 0x3 | revision  Identifies the revision:  0x0 for r0p0  0x1 for r1p0  0x2 for r1p1  0x3 for r1p2 |
| 3:0 | RO | 0x4 | designer\_1  Returns 0x4 |

Peripheral identification registers (periph\_id\_3)

Address: periph\_id\_3 + offset (0xfec)

Table 459.Peripheral identification registers (periph\_id\_3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | RO | 0x0 | Reserved, read undefined. |
| 7:1 | RO | 0x0 | Reserved for future use, read undefined. |
| 0 | RO | 0x0 | integration\_cfg  Returns 0 to indicate that the DMAC does not contain integration test logic |

Component identification registers 0-3 (pcell\_id\_ [3:0])

Address: pcell\_id\_ [3:0] + offset (0xff0~0xffc)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RO | 0xb1 | Returns 0xb1 |
| 23:16 | RO | 0x05 | Returns 0x05 |
| 15:8 | RO | 0xf0 | Returns 0xf0 |
| 7:0 | RO | 0x0d | Returns 0x0d |

* 1. Programming Sequence

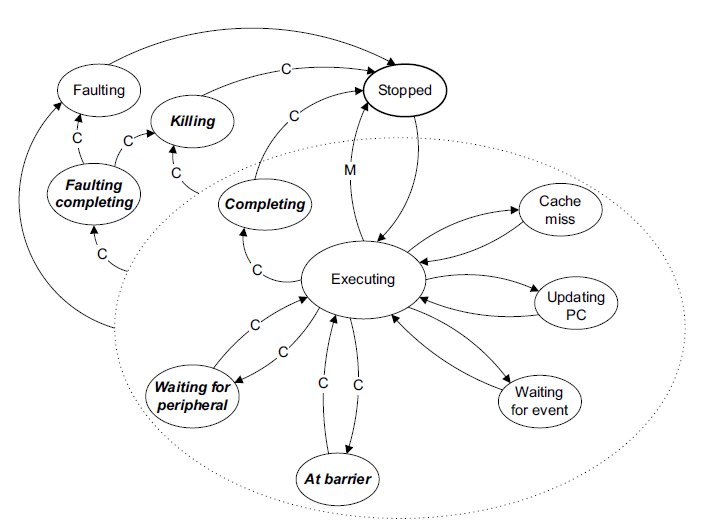


Figure 11.Thread operating states

In Figure 10, the DMAC permits that:

* + Only DMA channel threads can use states in bold italics
  + Arcs with no letter designator indicate state transitions for the DMA manager and DMA channel threads, otherwise use is restricted as follows:
* C DMA channel threads only.
* M DMA manager thread only.
* States within the dotted line can transition to the Faulting completing, Faulting, or Killing states.

After the DMAC exits from reset it sets all DMA channel threads to the Stopped state and the status of boot\_from\_pc controls the DMA manager thread state:

**Boot\_from\_pc is LOW**

DMA manager thread moves to the Stopped state.

**Boot\_from\_pc is HIGH**

DMA manager thread moves to the Executing state.

* + Stopped

The thread has an invalid PC and it is not fetching instructions. Depending on the thread type, you can cause the thread to move to the Executing state by:

* + DMA manager thread

With boot\_from\_pc HIGH and aresetn LOW then the DMA manager thread moves to the Executing state after aresetn goes HIGH.

* + DMA channel thread

Programming the DMA manager thread to execute DMAGO for a DMA channel thread in the Stopped state

* + Executing

The thread has a valid PC and therefore the DMAC includes the thread when it arbitrates. The thread can then change to one of the following states under the following conditions:

* + Stopped

When the DMA manager thread executes DMAEND

* + Cache miss

When the instruction cache does not contain the next instruction for either the DMA manager thread or the DMA channel thread

* + Updating PC

When the DMAC calculates the address of the next access in the cache

* + Waiting for event

When a thread executes DMAWFE

* + At barrier

When a DMA channel thread either:

* Executes DMARMB, DMAWMB, or DMAFLUSHP.
* Updates control registers that affect alignment
  + Waiting for peripheral

When a DMA channel thread executes DMAWFP

* + Killing

When a DMA channel thread executes DMAKILL

* + Faulting completing

For a DMA channel thread when either:

* The thread executes an undefined or invalid instruction.
* An AXI bus error occurs during an instruction fetch or data transfer.
  + Faulting

For the DMA manager thread when either:

* The thread executes an undefined or invalid instruction.
* An AXI bus error occurs during an instruction fetch.

For a DMA channel thread when a watchdog timeout abort occurs.

* + Completing

When a DMA channel thread executes DMAEND

* + Cache miss

The thread is stalled and the DMAC is performing a cache line fill. After it completes the cache fill, the thread returns to the Executing state

* + Updating PC

The DMAC is calculating the address of the next access in the cache. After it calculates the PC,the thread returns to the Executing state.

* + Waiting for event

The thread is stalled and is waiting for the DMAC to execute DMASEV using the corresponding event number. After the corresponding event occurs, the thread returns to the Executing state.

* + At barrier

A DMA channel thread is stalled and the DMAC is waiting for transactions on the AXI bus to complete. After the AXI transactions complete, the thread returns to the Executing state.

* + Waiting for peripheral

A DMA channel thread is stalled and the DMAC is waiting for the peripheral to provide the requested data. After the peripheral provides the data, the thread returns to the Executing state.

* + Faulting completing

A DMA channel thread is waiting for the AXI master interface to signal that the outstanding load or store transactions are complete. After the transactions complete, the thread moves to the Faulting state.

* + Faulting

The thread is stalled indefinitely. The thread moves to the Stopped state when you use the DBGCMD Register to instruct the DMAC to execute DMAKILL for that thread.( See Debug Command Register).

* + Killing

A DMA channel thread is waiting for the AXI master interface to signal that the outstanding load or store transactions are complete. After the transactions complete, the thread moves to the Stopped state.

* + Completing

A DMA channel thread is waiting for the AXI master interface to signal that the outstanding load or store transactions are complete. After the transactions complete, the thread moves to the Stopped state.

* 1. Programming Restrictions
     1. Fixed unaligned bursts

The DMAC does not support fixed unaligned bursts. If you program the following conditions, the DMAC treats this as a programming error:

* Unaligned read

The src\_inc filed is 0 in the CCRn Register, see Channel Control Registers.

The SARn Register contains an address that is not aligned to the size of data that the src\_burst\_size field contains, see Source Address Registers.

* Unaligned write

The dst\_inc field is 0 in the CCRn Register, see Channel Control Register.

The DARn Register contains an address that is not aligned to the size of data that the dst\_burst\_size field contains, see Destination Address Register.

* + 1. Endian swap size restrictions

If you program the endian\_swap\_size field in the CCRn Register, to enable a DMA channel to perform an endian swap, then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the size that the endian\_swap\_size field specifies before executing any DMALD or DMAST instructions.

If you update any of endian\_swap\_size, SARn, or DARn, for example, using a DMAADDH SAR instruction, then you must ensure that the SARn and DARn registers contain an address aligned to the size that the endian\_swap\_size field specifies before executing any additional DMALD or DMAST instructions.

If you program the src\_inc field in the CCRn Register to use a fixed address, you must program the src\_burst\_size field to select a burst size that is greater than or equal to the value that the endian\_swap\_size field specifies. Similarly, if you program the dst\_inc field to select a fixed destination address, you must program the dst\_burst\_size field to select a burst size that is greater than or equal to the value that the endian\_swap\_size field specifies.

If you program the dst\_inc field in the CCRn Register to use an incrementing address, you must program the CCRn Register so that dst\_burst\_len×dst\_burst\_size is a multiple of endian\_swap\_size. For example, if endian\_swap\_size = 0b010, 32-bit, and dst\_burst\_size = 0b001, 2 bytes per beat, then you can program dst\_burst\_len = 0b0001, 0b0011, 0b0101, ..., 0b1111, that is 2, 4, 6, ..., 16 transfers.

* + 1. Updating DMA channel control registers during a DMA cycle

Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address.

You can update these registers during a DMA cycle but if you change certain register fields then it can cause the DMAC to discard data. The following sections describe the register fields that might have a detrimental impact on a data transfer:

* Updates that affect the destination address
* Updates that affect the source address

Updates that affect the destination address

If you use a DMAMOV instruction to update the DARn Register or CCRn Register part way through a DMA cycle then this might cause a discontinuity in the destination data stream.

A discontinuity occurs if you change any of the following:

* endian\_swap\_size field
* dst\_inc bit
* dst\_burst\_size field when dst\_inc = 0,that is,fixed-address burst.DARn Register so that it modifies the destination byte lane alignment.For example, when the bus width is 64 bits and you change bits [2:0] in the DARn Register

When a discontinuity in the destination data stream occurs, the DMAC:

* Halts execution of the DMA channel thread.

Completes all outstanding read and write operations for the channel. That is, as if the DMAC was executing DMARMB and DMAWMB instructions.

Discards any residual MFIFO data for the channel

Resumes execution of the DMA channel thread

Updates that affect the source address

If you use a DMAMOV instruction to update the SARn Register or CCRn Register part way through a DMA cycle then this might cause a discontinuity in the source data stream.

A discontinuity occurs if you change any of the following:

* src\_inc bit
* src\_burst\_size field
* SARn Register so that it modifies the source byte lane alignment. For example, when the bus width is 32 bits and you change bits [1:0] in the SARn Register.
* When a discontinuity in the source data stream occurs, the DMAC:
* Halts execution of the DMA channel thread.
* Completes all outstanding read operations for the channel. That is, as if the DMAC was executing DMARMB instruction.
* Resumes execution of the DMA channel thread. No data is discarded from the MFIFO.

Resource sharing between DMA channels

DMA channel programs share the MFIFO data storage resource. You must not start a set of concurrently running DMA channel programs with a resource requirement that exceeds the configured size of the MFIFO.

The DMAC includes a mechanism called the load-lock to ensure that the shared MFIFO resource is used correctly. The load-lock is either owned by one channel, or it is free. The channel that owns the load-lock can execute DMALD instructions successfully. A channel that does not own the load-lock pauses at a DMALD instruction until it takes ownership of the load-lock.

A channel claims ownership of the load lock when:

* It executes a DMALD or DMALDP instruction.
* No other channel currently owns the load-lock.
* A channel releases ownership of the load-lock when any of the following occur:
* It executes a DMAST, DMASTP, or DMASTZ.
* It reaches a barrier, that is, it executes DMARMB or DMAWMB.
* It waits, that is, it executes DMAWFP or DMAWFE.
* It terminates normally, that is, it executes DMAEND.
* It aborts for any reason, including DMAKILL.

The MFIFO resource usage of a DMA channel program is measured in MFIFO entries, and rises and falls as the program proceeds. The MFIFO resource requirement of a DMA channel program is described using a static requirement and a dynamic requirement that are affected by the load-lock mechanism.

ARM defines the static requirement to be the maximum number of MFIFO entries that a channel is currently using before that channel does one of the following:

* Executes a WFP or WFE instruction.
* Claims ownership of the load-lock.

ARM defines the dynamic requirement to be the difference between the static requirement andthe maximum numbers of MFIFO entries that a channel program uses at any time during its execution.

To calculate the total MFIFO requirement, add the largest dynamic requirement to the sum of all the static requirements.

To avoid DMAC lock-up, the total MFIFO requirement of the set of channel programs must be equal to or less than the configured MFIFO depth.

Instruction for DMA

The DMAC instructions:

* Use a DMA prefix, to provide a unique name-space
* Have 8-bit opcodes that might use a variable data payload of 0, 8, 16, or 32 bits
* Use suffixes that are consistent.
  + 1. Instructions set summary

Table 460.Instruction syntax summary

|  |  |  |  |
| --- | --- | --- | --- |
| Mnemonic | Instruction | Thread usage: M = DMA manager C = DMA channel | |
| DMAADDH | Add Halfword | - | C |
| DMAADNH | Add Negative Halfword | - | C |
| DMAEND | End | M | C |
| DMAFLUSHP | Flush and Notify Peripheral | - | C |
| DMAGO | Go | M | - |
| DMAKILL | Kill | M | C |
| DMALD | Load | - | C |
| DMALDP | Load and Notify Peripheral | - | C |
| DMALP | Loop | - | C |
| DMALPEND | Loop End | - | C |
| DMALPFE | Loop Forever | - | C |
| DMAMOV | Move | - | C |
| DMANOP | No operation | M | C |
| DMARMB | Read Memory Barrier | - | C |
| DMASEV | Send Event | M | C |
| DMAST | Store | - | C |
| DMASTP | Store and Notify Peripheral | - | C |
| DMASTZ | Store Zero | - | C |
| DMAWFE | Wait For Event | M | C |
| DMAFP | Wait For Peripheral | - | C |
| DMAWMB | Write Memory Barrier | - | C |

* + 1. Instructions

The following sections describe the instructions that a DMAC can execute.

* + - 1. DMAADDH

Add Halfword adds an immediate 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMAC to support 2D DMA operations.

The immediate unsigned 16-bit value is zero-extended before the DMAC adds it to the address,using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. Figure 4 shows the instruction encoding.

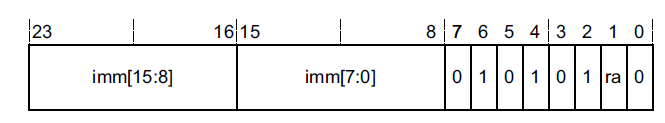


Figure 12.DMAADDH encoding

**Assembler syntax:** DMAADDH[address\_register], [16-bit immediate]

**[**address\_register**]** Selects the address register to use. It must be either:

**SAR** SARn Register and sets ra to 0.

**DAR** DARn Register and sets ra to 1.

**[**16-bit immediate**]** The immediate value to be added to the [address\_register].

Operation:You can only use this instruction in a DMA channel thread.

* + - 1. DMAADNH

Add Negative Halfword adds an immediate negative 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMAC to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses.

The immediate unsigned 16-bit value is one-extended to 32 bits, to create a value that is the two’s complement representation of a negative number between -65536 and -1, before the DMAC adds it to the address using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the current value in the Source or Destination Address Register.

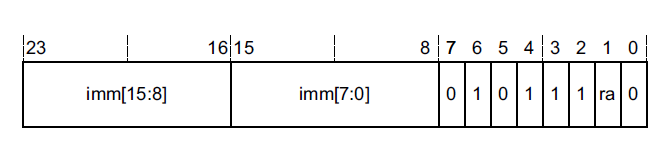


Figure 13.DMAADNH encoding

**Assembler syntax:** DMAADNH [address\_register], [16-bit immediate]

**[**address\_register**]** Selects the address register to use. It must be either:

**SAR** SARn Register and sets ra to 0.

**DAR** DARn Register and sets ra to 1.

**[**16-bit immediate**]** The immediate value to be added to the [address\_register].

Note: You must specify the 16-bit immediate as the number that is to be represented in the instruction encoding. For example, DMAADNH DAR, 0xFFF0 causes the value 0xFFFFFFF0 to be added to the current value of the Destination Address Register, effectively subtracting 16 from the DAR.

Operation: You can only use this instruction in a DMA channel thread.

* + - 1. DMAEND

End signals to the DMAC that the DMA sequence is complete. After all DMA transfers are complete for the DMA channel, the DMAC moves the channel to the Stopped state. It also flushes data from the MFIFO and invalidates all cache entries for the thread.

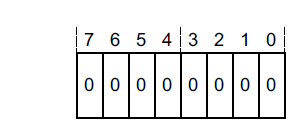


Figure 14.DMAEND encoding

**Assembler syntax：** DMAEND

Operation: You can use the instruction with the DMA manager thread and the DMA channel thread.

* + - 1. DMAFLUSHP

Flush Peripheral clears the state in the DMAC that describes the contents of the peripheral and sends a message to the peripheral to resend its level status.

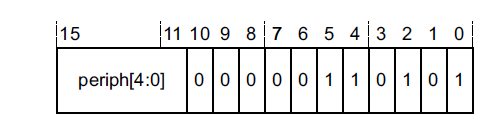


Figure 15.DMAFLUSHP encoding

**Assembler syntax：**DMAFLUSHP [peripheral]

[peripheral] 5-bit immediate, value 0-31.

Operation:You can only use this instruction in a DMA channel thread.

* + - 1. DMAGO

When the DMA manager executes Go for a DMA channel that is in the Stopped state, it performs the following steps on the DMA channel:

* Moves a 32-bit immediate into the program counter.
* Sets its security state.
* Updates it to the Executing state.

If a DMA channel is not in the Stopped state when the DMA manager executes DMAGO then the DMAC does not execute DMAGO but instead it executes DMANOP.

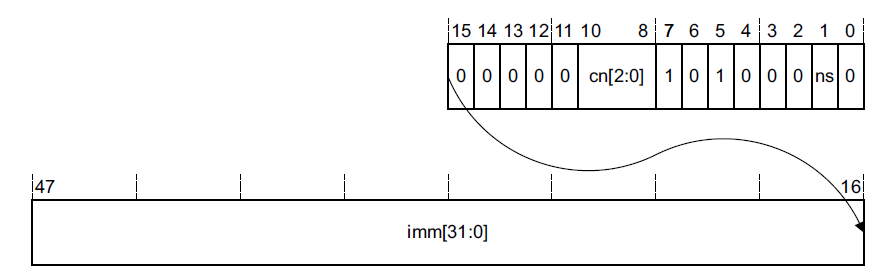


Figure 16.DMAGO encoding

**Assembler syntax：**DMAGO [channel\_number], [32-bit\_immediate] , [ns]

[channel\_number] Selects a DMA channel. It must be one of:

C0 DMA channel 0.

C1 DMA channel 1.

C2 DMA channel 2.

C3 DMA channel 3.

C4 DMA channel 4.

C5 DMA channel 5.

C6 DMA channel 6.

C7 DMA channel 7.

Note: If you provide a channel number that is not available for your configuration of the DMAC, the DMA manager thread abort.

[32-bit\_immediate] The immediate value that is written to the CPCn Register for the selected [channel\_number].

[ns] If ns is present, the DMA channel operates in the Non-secure state. Otherwise, the execution of the instruction depends on the security state of the DMA manager:

DMA manager is in the secure state

DMA channel operates in the secure state.

DMA manager is in the Non-secure state

The DMAC aborts.

Operation:You can only use this instruction with the DMA manager thread.

* + - 1. DMAKILL

Kill instructs the DMAC to immediately terminate execution of a thread. Depending on the thread type, the DMAC performs the following steps:

**DMA manager thread**

1. Invalidates all cache entries for the DMA manager.
2. Moves the DMA manager to the Stopped state.

**DMA channel thread**

1. Moves the DMA channel to the Killing state.
2. Waits for AXI transactions, with an ID equal to the DMA channel number, to complete.
3. Invalidates all cache entries for the DMA channel.
4. Remove all entries in the MFIFO for the DMA channel.
5. Remove all entries in the read buffer queue and write buffer queue for the DMA channel.
6. Moves the DMA channel to the Stopped state.

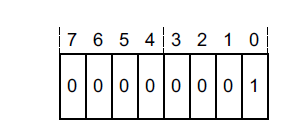


Figure 17.DMAKILL encoding

**Assembler syntax:** DMAKILL

Operation:You can use the instruction with the DMA manager thread and the DMA channel thread.

Note: You must not use the DMAKILL instruction in DMA channel programs. To issue a DMAKILL instruction, use the DBGINST0 Register.

* + - 1. DMALD[S|B]

Load instructs the DMAC to perform a DMA load, using AXI transactions. It places the read data into the MFIFO and tags it with the corresponding channel number. DMALD is an unconditional instruction but DMALDS and DMALDB are conditional on the state of the request\_type flag. If the src\_inc bit in the Channel Control Registers is set to incrementing, the DMAC updates the Source Address Registers after it executes DMALD[S|B].

Note: The DMAC sets the value of request\_type when it executes a DMAWFP instruction.

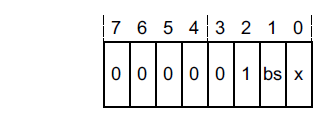


Figure 18.DMALD encoding

**Assembler syntax:** DMALD[S|B]

[S] If S is present, the assembler sets bs to 0 and x to 1. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMALD instruction and it sets arlen[3:0]=0x0 so that the AXI read transaction length is one. The DMAC ignores the value of the src\_burst\_len field in the Channel Control Registers.

request\_type = **Burst**

The DMAC performs a DMANOP instruction. The DMAC increments the channel PC to the next instruction. No state change occurs.

[B] If B is present, the assembler sets bs to 1 and x to 1. The instruction is conditional on

the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMANOP instruction. The DMAC increments the channel PC to the next instruction. No state change occurs.

request\_type = **Burst**

The DMAC performs a DMALD instruction. If you do not specify the S or B operand, the assembler sets bs to 0 and x to 0, and the DMAC always executes a DMA load.

Operation: You can only use this instruction in a DMA channel thread. If you specify the S or B operand, execution of the instruction is conditional on the state of request\_type matching that of the instruction.

* + - 1. DMALDP[S|B]

Load and notify Peripheral instructs the DMAC to perform a DMA load, using AXI transactions that Source Address Registers and Channel Control Registers. It places the read data into a FIFO that is tagged with the corresponding channel number and after it receives the last data item, it updates datype[1:0] to indicate to the peripheral that the data transfer is complete. If the src\_inc bit in the Channel Control Registers is set to incrementing, the DMAC updates Source Address Registers on page 3-23 after it executes DMALDP[S|B].

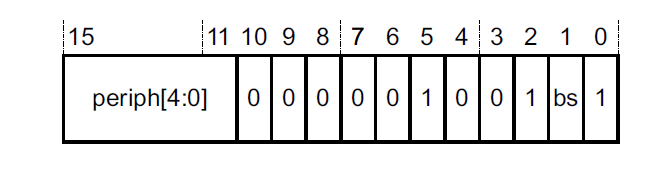


Figure 19. DMALDP[S|B] encoding

**Assembler syntax:** DMALDP[S|B] [peripheral]

[S] When S is present, the assembler sets bs to 0. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMALDP instruction and it sets arlen[3:0]=0x0 so that the AXI read transaction length is one. The DMAC ignores the value of the src\_burst\_len field in the Channel Control Registers.

request\_type = **Burst**

The DMAC performs a DMANOP.

[B] When B is present, the assembler sets bs to 1. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMANOP.

request\_type = **Burst**

The DMAC performs a load using a burst DMA transfer.

[peripheral] 5-bit immediate, value 0-31.

Note: The DMAC sets the value of the request\_type flag when it executes a DMAWFP instruction.

Operation:You can only use this instruction in a DMA channel thread. Execution of the instruction is conditional on the state of the request\_type flag matching that of the instruction

* + - 1. DMALP

Loop instructs the DMAC to load an 8-bit value into the Loop Counter Register you specify. This instruction indicates the start of a section of instructions, and you set the end of the section using the DMALPEND instruction. See DMALPEND[S|B] on page 4-11. The DMAC repeats the set of instructions that you insert between DMALP and DMALPEND until the value in the Loop Counter Register reaches zero.

Note: The DMAC saves the value of the PC for the instruction that follows DMALP. After the DMAC executes DMALPEND, and the Loop Counter Register is not zero, this enables it to execute the first instruction in the loop.

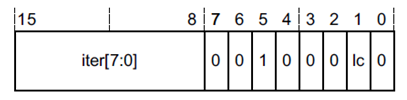


Figure 20.DMALP encoding

**Assembler syntax:** DMALP [loop\_iterations]

[loop\_iterations] Specifies the number of loops to perform, range 1-256.

Note: The assembler determines the Loop Counter Register to use and either:

Sets lc to 0, and the DMAC writes the value loop\_iterations minus 1 to the Loop Counter 0 Registers.

Sets lc to 1, and the DMAC writes the value loop\_iterations minus 1 to the Loop Counter 1 Registers.

Operation: You can only use this instruction in a DMA channel thread.

* + - 1. DMALPEND[S|B]

Loop End indicates the last instruction in the program loop but the behavior of the DMAC depends on whether DMALP or DMALPFE starts the loop. If a loop starts with:

DMALPThe loop has a defined loop count and DMALPEND[S|B] instructs the DMAC to read the value of the Loop Counter Register. If a Loop Counter Register returns:

**Zero** The DMAC executes a DMANOP and therefore exits the loop.

**Non-zero** The DMAC decrements the value in the Loop Counter Register and updates the thread PC to contain the address of the first instruction in the program loop, that is, the instruction that follows the DMALP.

**DMALPFE** The loop has an undefined loop count and the DMAC uses the state of therequest\_last flag to control when it exits the loop. If the request\_last flag is:

1. The DMAC updates the thread PC to contain the address of the first instruction in the program loop, that is, the instruction that follows the DMALP.
2. The DMAC executes a DMANOP and therefore exits the loop.

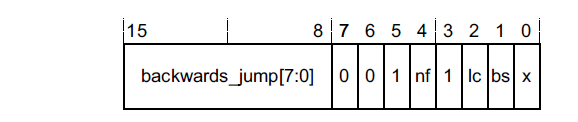


Figure 21.DMALPEND encoding

**Assembler syntax:** DMALPEND[S|B]

[S] If S is present and the loop starts with DMALP, then the assembler sets bs to 0 and x to 1. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC executes the DMALPEND instruction.

request\_type = **Burst**

The DMAC performs a DMANOP and therefore exits the loop.

[B] If B is present and the loop starts with DMALP, then the assembler sets bs to 1 and x to 1. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMANOP and therefore exits the loop.

request\_type = **Burst**

The DMAC executes the DMALPEND instruction.

If you do not specify the S or B operand, the assembler sets bs to 0 and x to 0, and the DMAC always executes the DMALPEND instruction.

Note: You must not specify the S or B operand when a loop starts with DMALPFE. If you do, the assembler issues a warning message and sets bs to 0, x to 0, and nf to 1. In the same way as for DMALPFE, the DMAC uses the state of the request\_last flag to control when it exits the loop.

The DMAC sets the value of the: request\_type flag when it executes a DMAWFP instruction. request\_last flag to 1 when the corresponding peripheral sets drlast HIGH, to signal the last peripheral request.

To correctly assign the additional bits in the DMALPEND instruction, that Figure 19 shows, the assembler determines the values for:

backwards\_jump[7:0] Sets the relative location of the first instruction in the program loop. The assembler calculates the value for backwards\_jump[7:0] by subtracting the address of the first instruction in the loop from the address of the DMALPEND instruction.

nf Sets it to:

0 if DMALPFE started the program loop.

1 if DMALP started the program loop.

lc Sets it to:

0 if the Loop Counter 0 Registers contains the loop counter value.

1 if the Loop Counter 1 Registers contains the loop counter value.

Operation: You can only use this instruction in a DMA channel thread. If you specify the S or B operand, execution of the instruction is conditional on the state of the request\_type flag matching that of the instruction.

* + - 1. DMALPFE

The assembler uses Loop Forever to configure certain bits in DMALPEND.

Note: When the assembler encounters DMALPFE, it does not create an instruction for the DMAC, but instead, it modifies the behavior of DMALPEND. The insertion of DMALPFE in program code identifies the start of the loop.

**Assembler syntax:** DMALPFE

* + - 1. DMAMOV

Move instructs the DMAC to move a 32-bit immediate into the following registers:

* Source Address Registers.
* Destination Address Registers.
* Channel Control Registers.

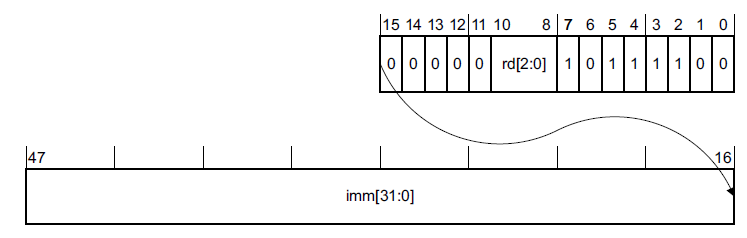


Figure 22.DMAMOV encoding

**Assembler syntax:** DMAMOV [destination\_register], [32-bit\_immediate]

[destination\_register] The valid registers are:

**SAR** Selects the Source Address Registers on page 3-23 and sets rd to 0b000.

**CCR** Selects the Channel Control Registers on page 3-25 and sets rd to 0b001.

**DAR** Selects the Destination Address Registers on page 3-24 and sets rd to 0b010.

[32-bit\_immediate] A 32-bit value that is written to the specified destination register.

Note: See DMAMOV CCR on page 4-22 for information about using the assembler to program the various fields that the Channel Control Registers.

Operation:You can only use this instruction in a DMA channel thread.

* + - 1. DMANOP

No Operation does nothing. You can use this instruction for code alignment purposes.

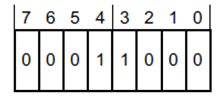


Figure 23.DMANOP encoding

**Assembler syntax:** DMANOP

Operation: You can use the instruction with the DMA manager thread and the DMA channel thread.

* + - 1. DMARMB

Read Memory Barrier forces the DMA channel to wait until all of the executed DMALD instructions for that channel have been issued on the AXI master interface and have completed. This enables write-after-read sequences to the same address location with no hazards.

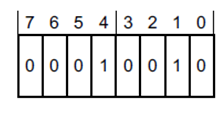


Figure 24.DMARMB encoding

**Assembler syntax:** DMARMB

Operation**:** You can only use this instruction in a DMA channel thread.

* + - 1. DMASEV

Send Event instructs the DMAC to modify an event-interrupt resource. Depending on how you program the Interrupt Enable Register, this either:

Generates event [event\_num].

Note: Typically, you use DMAWFE to stall a thread and then another thread executes DMASEV, using the appropriate event number, to unstall the waiting thread.

Signals an interrupt using irq[event\_num].

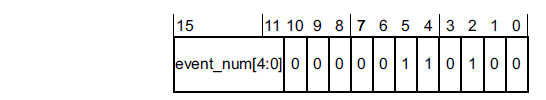


Figure 25.DMASEV encoding

**Assembler syntax:** DMASEV [event\_num]

[event\_num] 5-bit immediate, value 0-31.

Note: The DMAC aborts the thread if you select an event\_num that is not available for your configuration of the DMAC.

Operation:You can use the instruction with the DMA manager thread and the DMA channel thread. See Using events and interrupts for more information.

* + - 1. DMAST[S|B]

Store instructs the DMAC to transfer data from the FIFO to the location that the Destination Address registers, using AXI transactions that the DA Register and Channel Control Registers. If the dst\_inc bit in the Channel Control Registers is set to incrementing, the DMAC updates the Destination Address Registers after it executes DMAST [S|B].

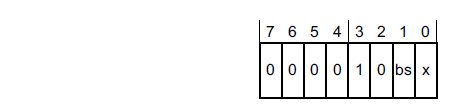


Figure 26.DMAST encoding

**Assembler syntax:** DMAST [S|B]

[S] If S is present, the assembler sets bs to 0 and x to 1. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMAST instruction and it sets len[3:0]=0x0 so that the AXI write transaction length is one. The DMAC ignores the value of the dst\_burst\_len field in the Channel Control Registers.

request\_type = **Burst**

The DMAC performs a DMANOP instruction. The DMAC increments the channel PC to the next instruction. No state change occurs.

[B] If B is present, the assembler sets bs to 1 and x to 1. The instruction is conditional on the state of the request\_type flag:

request\_type = **Single**

The DMAC performs a DMANOP instruction. The DMAC increments the channel PC to the next instruction. No state change occurs.

request\_type = **Burst**

The DMAC performs a DMAST instruction.If you do not specify the S or B operand, the assembler sets bs to 0 and x to 0, and the DMAC always executes a DMA store.

Note: The DMAC sets the value of the request\_type flag when it executes a DMAWFP instruction.

Operation: You can only use this instruction in a DMA channel thread. If you specify the S or B operand, execution of the instruction is conditional on the state of the request\_type flag matching that of the instruction.

The DMAC only commences the burst when the MFIFO contains all of the data necessary to complete the burst transfer.

* + - 1. DMASTP[S|B]

Store and notify Peripheral instructs the DMAC to transfer data from the FIFO to the location that the Destination Address Registers, using AXI transactions that the DA Register and Channel Control Registers. It uses the DMA channel number to access the appropriate location in the FIFO. After the DMA store is complete, and the DMAC has received a buffered write response, it updates datype[1:0] to notify the peripheral that the data transfer is complete. If the dst\_inc bit in the Channel Control Registers is set to incrementing, the DMAC updates the Destination Address Registers after it executes DMASTP[S|B].

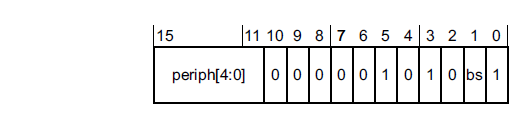


Figure 27.DMASTP encoding

**Assembler syntax:** DMASTP [S|B] [peripheral]

**[S]** Sets bs to 0. This instructs the DMAC to perform:

A single DMA store operation if request\_type is programmed to Single.

Note:The DMAC ignores the state of the dst\_burst\_len field in the Channel Control Registers on page 3-25 and always performs an AXI transfer with a burst length of one. A DMANOP if request\_type is programmed to Burst.

**[B]**  Sets bs to 1. This instructs the DMAC to perform:

The DMA store if request\_type is programmed to Burst.

A DMANOP if request\_type is programmed to Single.

**[peripheral]** 5-bit immediate, value 0-31.

Note: The DMAC sets the value of the request\_type flag when it executes a DMAWFP instruction.

Operation: You can only use this instruction in a DMA channel thread.The DMAC only commences the burst when the MFIFO contains all of the data necessary to complete the burst transfer.

* + - 1. DMASTZ

Store Zero instructs the DMAC to store zeros, using AXI transactions that the Destination Address Registers and Channel Control Registers. If the dst\_inc bit in the Channel Control Registers is set to incrementing, the DMAC updates the Destination Address Registers after it executes DMASTZ.

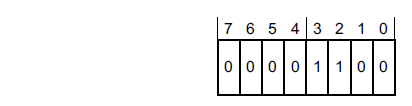


Figure 28.DMASTZ encoding

**Assembler syntax:** DMASTZ

Operation: You can only use this instruction in a DMA channel thread.

* + - 1. DMAWFE

Wait For Event instructs the DMAC to halt execution of the thread until the event, that event\_num specifies, occurs. When the event occurs, the thread moves to the Executing state and the DMAC clears the event.

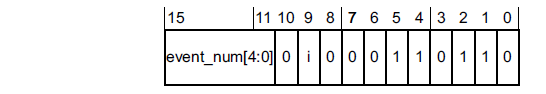


Figure 29.DMAWFE encoding

**Assembler syntax:** DMAWFE [event\_num], [ invalid]

**[event\_num]** 5-bit immediate, value 0-31.

**[invalid]** Sets i to 1. If invalid is present, the DMAC invalidates the instruction cache for the current DMA thread. If invalid is not present, then the assembler sets i to 0 and the DMAC does not invalidate the instruction cache for the current DMA thread.

Note: The DMAC aborts the thread if you select an event\_num that is not available for your configuration of the DMAC. To ensure cache coherency, you must use invalid when a processor writes the instruction stream for a DMA channel.

Operation: You can use the instruction with the DMA manager thread and the DMA channel thread.

* + - 1. DMAWFP

Wait For Peripheral instructs the DMAC to halt execution of the thread until the specified peripheral signals a DMA request for that DMA channel.

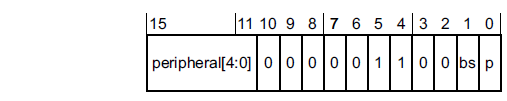


Figure 30.DMAWFP encoding

**Assembler syntax:** DMAWFP [peripheral], [single|burst|periph]

**[peripheral]** 5-bit immediate, value 0-31.

Note: The DMAC aborts the thread if you select a peripheral number that is not available you’re your configuration of the DMAC.

**[single]** Sets bs to 0 and p to 0. This instructs the DMAC to continue executing the DMA channel thread after it receives a single or burst DMA request. The DMAC sets the request\_type to Single, for that DMA channel.

**[burst]** Sets bs to 1 and p to 0. This instructs the DMAC to continue executing the DMA channel thread after it receives a burst DMA request. The DMAC sets the request\_type to Burst.

Note: The DMAC ignores single burst DMA requests.

**[periph]** Sets bs to 0 and p to 1. This instructs the DMAC to continue executing the DMA channel thread after it receives a single or burst DMA request.

The DMAC sets the request\_type to:

**Single** When it receives a single DMA request.

**Burst**  When it receives a burst DMA request.

Operation:You can only use this instruction in a DMA channel thread.

* + - 1. DMAWMB

Write Memory Barrier forces the DMA channel to wait until all of the executed DMAST instructions for that channel have been issued on the AXI master interface and have completed. This permits read-after-write sequences to the same address location with no hazards.

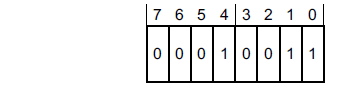


Figure 31.DMAWMB encoding

**Assembler syntax:** DMAWMB

Operation: You can only use this instruction in a DMA channel thread.

1. GMAC Controller
   1. Register Summary

The Control and Status Registers are located physically on the MAC-1G/MAC core and can be accessed directly by a host. All the CSRs are 32 bits long and quad word-aligned.

Table 461.GMAC Controller Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| CSR0 | 0x000 | 0xFE000000 | Bus mode |
| CSR1 | 0x008 | 0x00000000 | Transmit poll demand |
| CSR2 | 0x010 | 0x00000000 | Receive poll demand |
| CSR3 | 0x018 | 0x00000000 | Receive list base address |
| CSR4 | 0x020 | 0x00000000 | Transmit list base address |
| CSR5 | 0x028 | 0x00000000 | Status |
| CSR6 | 0x030 | 0x32000040 | Operation mode |
| CSR7 | 0x038 | 0x00000000 | Interrupt enable |
| CSR8 | 0x040 | 0xE0000000 | Missed frames and overflow counters |
| CSR9 | 0x048 | 0x00000000 | Serial ROM / Software MII serial management |
| CSR10 | 0x050 | 0x30000000 | MII serial management |
| CSR11 | 0x058 | 0x00000000 | Timer and interrupt mitigation control |
| - | 0x060 | - | not implemented |
| - | 0x068 | - | not implemented |
| - | 0x070 | - | not implemented |
| - | 0x078 | - | not implemented |
| CSR16 | 0x080 | 0x00000000 | MAC address low |
| CSR17 | 0x088 | 0x00000000 | MAC address high |
| CSR18 | 0x090 | 0x00000000 | Pause time & cache thresholds |
| CSR19 | 0x098 | 0x00000000 | Pause control FIFO thresholds |
| CSR20 | 0x0A0 | 0x30000000 | Flow control setup & status |
| - | 0x0A8-0xF8 | - | not implemented |
| SC00 - SC63 | 0x200-0x3F8 | undefined | Statistical Counters |

* 1. Register Descriptions

The bus mode register CSR0

Address: Operational Base + offset (0x000)

Table 462.Bus mode register CSR0

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:21 | - | - | Reserved |
| 20 | RW | 0x0 | Descriptor byte ordering mode.  1 – big endian mode used for data descriptors.  0 – little endian mode used for data descriptors.  DBO can be written only when both the receive and the transmit processes are stopped. |
| 19:17 | RW | 0x0 | Transmit automatic polling.  If the TAP is written with a nonzero value, the MAC-1G/MAC performs an automatic transmit descriptor polling when operating in suspended state. When the descriptor is available, the transmit process goes into the running state. When the descriptor is marked as owned by the host, the transmit process remains suspended. The poll is always performed at the current transmit descriptor list position.  The time interval between two consecutive polls is shown in  Table 192. The actual interval is based on the network side (GMII /RMII/RGMII) clock source. It is calculated by dividing clkt clock. This produces different absolute time values for each operating mode (10/100/1000 ).  TAP can be written only when the transmit processes is stopped. |
| 13:8 | RW | 0x0 | Programmable burst length for data buffer access.  Specifies the maximum number of words that can be transferred within one DMA transaction. Values permissible are 0, 1, 2, 4, 8, 16, and 32. When a '0' value is written, the bursts are limited only by the internal FIFO's threshold levels and host bus burst possibilities\*  The width of the single word is equal to the DATAWIDTH generic parameter, i.e. all data transfers always use the maximum data bus width.  Note that the PBL is valid only for the data buffers. The data descriptors burst length depends on a DATAWIDTH parameter. The rule is that every descriptor is accessed with a single burst.  PBL can be written only when both the receive and the transmit processes are stopped.  the maximum burst length for OCP/AHB/AXI/Generic interface implementations is 63 words. |
| 7 | RW | 0x0 | Big/Little endian.  Selects the byte-ordering mode used by the data buffers.  1 – big endian mode used for the data buffers.  0 – little endian mode used for the data buffers.  BLE can be written only when both the receive and the transmit processes are stopped. |
| 6:2 | RW | 0x0 | Descriptors skip length.  Specifies the number of 128-bit words between two consecutive unchained descriptors.  For example, when the DSL=0 then consecutive descriptors are read from following addresses 0x\*\*\*00, 0x\*\*\*10, 0x\*\*\*20, etc; when the DSL=1 then consecutive descriptors are read from following addresses 0x\*\*\*00, 0x\*\*\*20, 0x\*\*\*40, etc;  DSL can be written only when both the receive and the transmit processes are stopped. |
| 1 | RW | 0x0 | Bus arbitration mode.  0 – means that the Rx process has priority over Tx process. When both Rx and Tx processes request the bus, than the sequence of transfers on the DMA bus looks like: Rx->Rx->Tx->Rx->Rx->Tx… In other words, single Tx access is placed between every two consecutive Rx accesses. Using this mode helps to prevent Rx FIFO from overflow.  1 – means that Rx and Tx processes share equal priority for DMA access. Round Robin arbitration is used in which the sequence of transfers on the DMA bus looks like: Rx->Tx->Rx->Tx->Rx… |
| 0 | RW | 0x0 | Software reset.  Setting this bit resets all internal flip-flops except external PHY interface wrappers and AHB interface wrappers. SWR will be cleared automatically when the reset operation is complete. |

Table 463.Transmit automatic polling intervals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CSR(19..17) | Time interval | 10Mbps | 100Mbps | 1Gbps |
| 000 | TAP disabled | - | - | - |
| 001 | (g)mii\_period \* 128 | 51.2 us | 5.12 us | 1.024 us |
| 010 | (g)mii\_period \* 256 | 102.4 us | 10.24 us | 2.048 us |
| 011 | (g)mii\_period \* 512 | 204.8 us | 20.48 us | 4.096 us |
| 100 | (g)mii\_period \* 1024 | 409.6 us | 40.96 us | 8.192 us |
| 101 | (g)mii\_period \* 2048 | 819.2 us | 81.92 us | 16.384 us |
| 110 | (g)mii\_period \* 4096 | 1.638 ms | 163.8 us | 32.768 us |
| 111 | (g)mii\_period \* 8192 | 3.276 ms | 327.6 us | 65.536 us |

The transmit poll demand register CSR1

Address: Operational Base + offset (0x008)

Table 464.The transmit poll demand register CSR1

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | Writing this field with any value instructs the MAC-1G/MAC to check for frames to be transmitted. This operation is valid only when the transmit process is suspended.  If no descriptor is available the transmit process remains suspended.  When the descriptor is available the transmit process enters the running state.  This is write only register, reading always returns 0x00000000 value. |

The receive poll demand register CSR2

Address: Operational Base + offset (0x010)

Table 465.The receive poll demand register CSR2

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | Writing this field with any value instructs the MAC-1G/MAC to check for receive descriptors to be acquired. This operation is valid only when the receive process is suspended.  If no descriptor is available the receive process remains suspended.  When the descriptor is available the receive process enters the running state.  This is writing only register, reading always returns 0x00000000 value |

The receive descriptor list base address register CSR3

Address: Operational Base + offset (0x018)

Table 466.The receive descriptor list base address register CSR3

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | Start of the receive list address.  Contains the address of the first descriptor in a receive descriptor list. This address should be 64-bit aligned (RLA(2..0)=000).  This register can be written only when the receive process is stopped. |

The transmit descriptor list base address register CSR4

Address: Operational Base + offset (0x020)

Table 467.The transmit descriptor list base address register CSR4

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | Start of the transmit list address.  Contains the address of the first descriptor in a transmit descriptor list. This address should be 64-bit aligned (TLA(2..0)=000).  This register can be written only when the transmit process is stopped. |

The status register CSR5

Address: Operational Base + offset (0x028)

Table 468.The status register CSR5

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:23 | - | - | Reserved |
| 22:20 | R | 0x0 | Transmit process state.  Indicates the current state of a transmit process:  000 - Stopped, RESET or STOP TRANSMIT command issued  001 - Running, fetching transmit descriptor.  010 - Running, waiting for end of transmission.  011 - Running, transferring data buffer from host memory to FIFO.  100 - Reserved.  101 - Running, setup packet processing.  110 - Suspended, FIFO underflow or unavailable descriptor.  111 – Running, closing transmit descriptor. |
| 19:17 | R | 0x0 | Receive process state .  Indicates the current state of a receive process:  000 - Stopped, RESET or STOP RECEIVE command issued.  001 - Running, fetching receive descriptor.  010 - Running, waiting for the end of receive packet.  011 - Running, waiting for receive packet.  100 - Suspended, unavailable receive buffer.  101 - Running, closing receive descriptor.  110 - Not used  111 – Running, transferring data from FIFO to host memory. |
| 16 | R | 0x0 | Normal interrupt summary.  This bit is logical or on the following bits:  CSR5.0 - Transmit interrupt  CSR5.2 - Transmit buffer unavailable  CSR5.6 - Receive interrupt  CSR5.11 - General purpose timer overflow  CSR5.14 – Early receive interrupt  Only the unmasked bits affect the normal interrupt summary bit. |
| 15 | R | 0x0 | Abnormal interrupt summary.  This bit is logical or on the following bits:  CSR5.1 - Transmit process stopped  CSR5.5 - Transmit underflow  CSR5.7 - Receive buffer unavailable  CSR5.8 - Receive process stopped  CSR5.10 – Early transmit interrupt  Only the unmasked bits affect the abnormal interrupt summary bit. |
| 14 | RW | 0x0 | Early receive interrupt.  Set when the MAC-1G/MAC fills the data buffers of the first descriptor. Cleared by the MAC-1G/MAC after the receive interrupt (CSR5.6).  The user can clear this bit by writing 1.  Writing 0 has no effect. |
| 13:12 | - | - | Rserved |
| 11 | RW | 0x0 | General-purpose timer expiration.  Set when the general-purpose timer reaches a value of zero.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 10 | RW | 0x0 | Early transmit interrupt.  Indicates that the packet to be transmitted was fully transferred into the FIFO. This bit is cleared by the MAC-1G after the transmit interrupt (CSR5.0).  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 9 | - | - | Reserved |
| 8 | RW | 0x0 | Receive process stopped.  RPS is set when a receive process enters a stopped state.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 7 | RW | 0x0 | Receive buffer unavailable.  When set, indicates that the next receive descriptor is owned by the host and is unavailable for the MAC-1G/MAC. When RU becomes set, the MAC-1G/MAC enters a suspended state, and returns to receive descriptor processing when the host changes ownership of the descriptor and either a receive poll demand command is issued or a new frame is recognized by the MAC-1G/MAC.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 6 | RW | 0x0 | Receive interrupt.  Indicates the end of a frame receive. The complete frame has been transferred into the receive buffers.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 5 | RW | 0x0 | Transmit underflow.  Indicates that the transmit FIFO was empty during a transmission. The transmit process goes into a suspended state.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 4:3 | - | - | Reserved |
| 2 |  | 0x0 | Transmit buffer unavailable.  When set, TU indicates that the host owns the next descriptor on the transmit descriptor list therefore it cannot be used by the MAC-1G/MAC. When the TU becomes set, the transmit process goes into a suspended state and can resume normal descriptor processing when the host changes ownership of the descriptor and either a transmit poll demand command is issued or transmit automatic polling is enabled.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 1 |  | 0x0 | Transmit process stopped.  TPS is set when the transmit process goes into a stopped state.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |
| 0 |  | 0x0 | Transmit interrupt.  Indicates the end of a frame transmission process.  The user can clear this bit by writing a '1'.  Writing a '0' has no effect. |

The operation mode register CSR6

Address: Operational Base + offset (0x030)

Table 469.The operation mode register CSR6

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | - | - | Reserved |
| 30 | RW | 0x0 | Receive all.  When set, all incoming frames are received regardless of their destination address. An address check is performed and the result of the check is written into the receive descriptor (RDES0.30).  RA can be changed only when the receive process is in a stopped state. |
| 29:23 | - | - | Reserved |
| 22 | RW | 0x0 | Transmit threshold mode.  TTM can be changed only when the transmit process is in a stopped state. |
| 21 | RW | 0x0 | Store and forward.  When set, the transmission starts after a full packet is written into the transmit FIFO, regardless of the current FIFO threshold level.  SF can be changed only when the transmit process is in the stopped state. |
| 20:18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Ethernet Speed Selection.  The value in this field is used to drive the external speed pin used to configure external components as defined below.   |  |  |  |  | | --- | --- | --- | --- | | CSR6.17 | CSR6.16 | Ethernet speed | speed[1:0] | | 1 | 0 | 10Mbps | 00 | | 0 | 0 | 100Mbps | 01 | | 0 | 1 | 1 Gbps | 10 | | 1 | 1 | 1 Gbps | 10 |   SPEED bits can be changed only when both the transmit and the receive processes are in the stopped state.  Note: Gigabit version o the core (MAC-1G) implements 2-bit wide SPEED field (CSR6.17..16) while 10/100Mbit/s version of the core (MAC) implements 1-bit SPEED field (CSR6.17). |
| 15:14 | RW | 0x0 | Threshold control bits.  This bit, together with the TTM, and the SF, control the threshold level for the transmit FIFO.  TR can be changed only when the transmit process is in a stopped state. |
| 13 | RW | 0x0 | Start/stop transmit command.  Setting this bit when the transmit process is in a stopped state causes transition into a running state. In the running state the MAC-1G/MAC checks the transmit descriptor at a current descriptor list position. If the MAC-1G/MAC owns the descriptor then the data starts to transfer from memory into the internal transmit FIFO.  When the host owns the descriptor, the MAC-1G/MAC enters a suspended state.  Clearing this bit when the transmit process is in a running or a suspended state instructs the MAC-1G/MAC to enter the stopped state. |
| 12:11 | - | - | Reserved |
| 10 | RW | 0x0 | Loopback mode.  0 – normal operation  1 – internal loopback.  When working in the internal loopback mode, the transmitted frames are routed back to the receiver inside the core. The external loopback pin should be used to switch the receive clock externally, on the chip-level. The system designer should provide the receive clock synchronous to the transmit clock when working in the loopback. Standard receive clock recovered from the PHY cannot be used in the loopback mode. |
| 9 | RW | 0x0 | Full duplex mode.  0 – half duplex mode.  1 – forcing full duplex mode.  Changing FD is allowed only when both the transmitter and receive processes are in the stopped state. |
| 8 | - | - | Reserved |
| 7 | RW | 0x0 | Pass all multicast.  When set, all the frames with the multicast destination addresses will be received regardless of the address check result.  PB can be changed only when the receive process is in a stopped state. |
| 6 | RW | 1’b1 | Promiscuous mode.  When set all the frames will be received regardless of the address check result. An address check is not performed.  PR can be changed only when the receive process is in a stopped state. |
| 5 | - | - | Reserved |
| 4 | R | 0x0 | Inverse filtering.  If this bit is set when working in a perfect filtering mode, the receiver performs an inverse filtering during the address check process.  The "filtering type" bits of the setup frame determine a state of this bit. |
| 3 | RW | 0x0 | Pass bad frames.  When set, the MAC-1G/MAC transfers all frames into the data buffers, regardless of the receive errors. This allows the runt frames, collided fragments and truncated frames to be received.  PB can be changed only when the receive process is in a stopped state. |
| 2 | R | 0x0 | Hash only filtering mode.  When set, the MAC-1G/MAC performs an imperfect filtering over both the multicast and the physical addresses.  The "filtering type" bits of the setup frame determine the state of this bit. |
| 1 | RW | 0x0 | Start/stop receive command.  Setting this bit when the receive process is in a stopped state causes the transition into a running state. In the running state the MAC-1G/MAC checks the receive descriptor at the current descriptor list position. If the MAC-1G/MAC owns the descriptor, then it can process an incoming frame. When the host owns the descriptor, the receiver enters a suspended state and also sets the CSR5.7 (receive buffer unavailable) bit.  Clearing this bit when the receive process is in running or suspended state instructs the MAC-1G/MAC to enter a stopped state after receiving the current frame. |
| 0 | R | 0x0 | Hash/perfect receive filtering mode.  0 – perfect filtering of the incoming frames is performed according to the physical addresses specified in a setup frame.  1 – imperfect filtering over the frames with the multicast addresses is performed according to the hash table specified in a setup frame.  A physical addresses check is performed according to the CSR6.2 (HO - Hash only) bit.  When the HO and HP are both set, an imperfect filtering is performed on all of the addresses.  The "filtering type" bits of the setup frame determine the state of this bit. |

The interrupt enable register CSR7

Address: Operational Base + offset (0x038)

Table 470.The interrupt enable register CSR7

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:17 | - | - | Reserved |
| 16 | RW | 0x0 | Normal interrupt summary enable. When set, normal interrupts are enabled.  Normal interrupts are listed below:  CSR5.0 – Transmit interrupt  CSR5.2 – Transmit buffer unavailable  CSR5.6 – Receive interrupt  CSR5.11 – General-purpose timer expired  CSR5.14 – Early receive interrupt |
| 15 | RW | 0x0 | Abnormal interrupt summary enable.  When set, abnormal interrupts are enabled.  Abnormal interrupts are listed below:  CSR5.1 – Transmit process stopped  CSR5.5 – Transmit underflow  CSR5.7 – Receive buffer unavailable  CSR5.8 – Receive process stopped  CSR5.10 – Early transmit interrupt |
| 14 | RW | 0x0 | Early receive interrupt enable.  When both the ERE and normal interrupt enable bits are set, early receive interrupt is enabled. |
| 13:12 | - | - | Reserved |
| 11 | RW | 0x0 | General-purpose timer overflow enable.  When both the GTE and normal interrupt summary enable bits are set, the general-purpose timer overflow interrupt is enabled. |
| 10 | RW | 0x0 | Early transmit interrupt enable.  When both the ETE and abnormal interrupt summary enable bits are set, the early transmit interrupt is enabled. |
| 9 | - | - | Reserved |
| 8 | RW | 0x0 | Receive stopped enable.  When both the RSE and abnormal interrupt summary enable bits are set, the receive stopped interrupt is enabled. |
| 7 | RW | 0x0 | Receive buffer unavailable enable.  When both the RUE and abnormal interrupt summary enable bits are set, the receive buffer unavailable is enabled. |
| 6 | RW | 0x0 | Receive interrupt enable.  When both the RIE and normal interrupt summary enable bits are set, the receive interrupt is enabled. |
| 5 | RW | 0x0 | Underflow interrupt enable.  When both the UNE and abnormal interrupt summary enable bits are set, the transmit underflow interrupt is enabled. |
| 4:3 | - | - | Reserved |
| 2 | RW | 0x0 | Transmit buffer unavailable enable.  When both the TUE and normal interrupt summary enable bits are set, the transmit buffer unavailable interrupt is enabled. |
| 1 | RW | 0x0 | Transmit stopped enable.  When both the TSE and abnormal interrupt summary enable bits are set, the transmit process stopped interrupt is enabled. |
| 0 | RW | 0x0 | Transmit interrupt enable.  When both the TIE and normal interrupt summary enable bits are set, the transmit interrupt is enabled. |

The missed frames counter register CSR8

Address: Operational Base + offset (0x040)

Table 471.The missed frames counter register CSR8

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:29 | - | - | Reserved |
| 28 | R | 0x0 | Overflow counter overflow (. Gets set when the FIFO overflow counter overflow. Reset when read. |
| 27:17 | R | 0x0 | FIFO overflow counter. Counts the number of frames truncated due to an overflow. Such frames are also reported to the receive descriptor status (RDES0.ZERO). Reset when read. |
| 16 | R | 0x0 | Missed frame overflow.  Set when a missed frame counter overflows. Reset when read. |
| 15:0 | R | 0x0 | Missed frame counter. Counts the number of frames not accepted due to the internal error of MAC-1G/MAC. The error can be caused by the FIFO overflow or by the receive descriptor unavailability. Reset when read. |

The serial ROM/Software MII Serial Management register CSR9

Address: Operational Base + offset (0x048)

Table 472.The serial ROM / Software MII Serial Management register CSR9

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:20 | - | - | Reserved |
| 19 | R | 0x0 | MII management data in signal.  This bit reflects the sample on the MDI pin during the read operation on the MII management interface. |
| 18 | RW | 0x0 | MII management interface mode  1 – indicates that the mdio pin is driven with the mdo value and data or command can be written to PHY  0 – indicates that the mdio pin is set to the high impedance state and the data can be read from PHY |
| 17 | RW | 0x0 | MII management write data.  The value of this bit drives the MDO pin when a write operation is performed of the MAC-1G/MAC. |
| 16 | RW | 0x0 | MII management clock.  The value of this bit drives the MDC pin of the MAC-1G/MAC. |
| 15:4 | - | - | Reserved |
| 3 | RW | 0x0 | Serial ROM data output.  The value of this bit drives the sdo pin of the MAC-1G/MAC. |
| 2 | R | 0x0 | Serial ROM data input (Read only).  This bit reflects the sdi pin of the MAC-1G/MAC. |
| 1 | RW | 0x0 | Serial ROM clock.  The value of this bit drives the sclk pin of the MAC-1G/MAC. |
| 0 | RW | 0x0 | Serial ROM chip select.  The value of this bit drives the scs pin of the MAC-1G/MAC. |

The MII serial management register CSR10

Address: Operational Base + offset (0x050)

Table 473.The MII serial management register CSR10

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | RW | 0x0 | Writing this bit with "1" starts management data transfer or applying MIISM settings respectively to used opcode. When read as "1" indicates that data transition / setting applying process is in progress; When read as "0" indicates data transition / setting applying process completion. |
| 30:28 | R | 3‘b011 | When register is read this field contains actual MIISM clock divider settings Writing register with new CLKDIV value simultaneously with OPCODE set to "11" and START bit set, applying new clock divider settings on next rising edge of MDCLK signal.  Clock divider settings:  "000" - MDC = clk / 8  "001" - MDC = clk / 16  "010" - MDC = clk / 32  "011" - MDC = clk / 64  "100" - MDC = clk / 128  "101" - MDC = clk / 256  "110" - MDC = clk / 512  "111" - MDC = clk / 1024  The default value is "011" and the management clock generation is disabled after reset. |
| 27:26 | RW | 0x0 | MIISM operation code:  "00" – disable clock generation,  "01" - register write command,  "10" - register read command,  "11" - clock divider set.  Applying commands “00” disables the management clock generation while applying any other command enables the management clock generation. |
| 25:21 | RW | 5‘b100000 | Physical layer address for current transfer |
| 20:16 | RW | 0x0 | Register address for the current transfer. |
| 15:0 | RW | 0x0 | Register data.  When write operation is selected (OPCODE="10"), the data from this filed will be written to selected register. When register read code is used (OPCODE="01"), the field will be overwritten by the data from selected register. |

The general-purpose timer and interrupt mitigation control register CSR11

Address: Operational Base + offset (0x058)

Table 474.The general-purpose timer and interrupt mitigation control register CSR11

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | RW | 0x0 | Cycle size. Controls the time units for the transmit and receive timers according to the following table:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | CS | Time interval | 10Mbps | 100Mbps | 1Gbps | | 1 | (g)mii\_period \* 128 | 51.2 us | 5.12 us | 1.024 us | | 0 | (g)mii\_period \* 2048 | 819.2 us | 81.92 us | 16.384 us | |
| 30:27 | R | 0x0 | Transmit timer. Controls the time that must elapse between the end of a transmit operation and issuing the transmit interrupt. This time is equal to TT \* (16\*CS). Writing 0 to this field disables the timer effect on the transmit interrupt mitigation mechanism.  When read, TT shows the actual timer value decremented on each 16th cycle defined by CS bit. |
| 26:24 | RW | 0x0 | Number of transmit packets.  Controls the number of the frames transmitted before issuing the transmit interrupt.  Writing 0 to this field disables the counter effect on the transmit interrupt mitigation mechanism.  When read, NTP shows the actual counter value decremented on each transmitted packet. |
| 23:20 | RW | 0x0 | Receive timer.  Controls the time that must elapse between the end of a receive operation and issuing the receive interrupt. This time is equal to RT \* CS.  Writing 0 to this field disables the timer effect on the receive interrupt mitigation mechanism.  When read, RT shows the actual timer value decremented on each cycle defined by CS bit. |
| 19:17 | RW | 0x0 | Number of receive packets.  Controls the number of the received frames before issuing the receive interrupt. Writing 0 to this field disables the timer effect on the receive interrupt mitigation mechanism.  When read, NRP shows the actual counter value decremented on each received packet. |
| 16 | RW | 0x0 | Continuous mode.  1 – general-purpose timer works in continuous mode.  0 – general-purpose timer works in one-shot mode. |
| 15:0 | RW | 0x0 | General purpose timer.  Contains the number of iterations of the general-purpose timer. Each iteration duration is defined by CSR11.31 (CS) bit. The entire general purpose timer period is equal to TIM \* CS.  When TIM is written with nonzero value, the timer starts to count down. After reaching 0, the behavior of the timer depends on CSR11.16 (CON) settings. In one-shot mode (CON=0), the timer stops. In continuous mode (CON=1), it is reloaded with the latest value written by the software to TIM register, and continues to count down. Interrupt CSR5.11 (GTI) can be generated whenever timer reaches 0 value.  When read, TIM shows the actual timer value. |

The MAC address low register CSR16

Address: Operational Base + offset (0x080)

Table 475.MAC address low register CSR16

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | MAC address low bytes. This field contains 4 lowest bytes of the MAC hardware address ( MAL(31..0) = MAC(31..0) ).  This field is used to fill the source hardware address field of pause frames generated by host. |

MAC address high register CSR17

Address: Operational Base + offset (0x088)

Table 476.MAC MAC address high register CSR17

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0000 | MAC address high bytes. This field contains 2 highest bytes of the MAC hardware address ( MAH(15..0) = MAC(63..32) ). This field is used to fill the source hardware address field of pause frames generated by host. |

Pause time & cache thresholds register CSR18

Address: Operational Base + offset (0x090)

Table 477.Pause time & cache thresholds register CSR18

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RW | 0x0 | Cache pause threshold level.  The filed contains unsigned value indicating the minimum number of frames stored in the receive FIFO when the flow control mechanism starts to transmit pause frames.  The actual width of the field depends of RCACHE\_DEPTH parameter and a maximum value is RCACHE\_DEPTH-1 (the significant bits are from 24 to 23+RCACHE\_DEPTH).  Writing a value less than in a CRTL field or 0 disables the flow control reactions on frame cash overflow.. |
| 23:16 | RW | 0x0 | Cache restart threshold level.  The filed contains unsigned value indicating the maximum number of frames stored in the receive FIFO below which the flow control mechanism stops to transmit pause frames.  The actual width of the field depends of RCACHE\_DEPTH parameter and a maximum value is RCACHE\_DEPTH-1 (the significant bits are from 16 to 15+RCACHE\_DEPTH).  Writing a value larger than in a CPTL field or 0 disables the flow control reactions on frame cash overflow. |
| 15:0 | RW | 0x0 | Flow control pause quanta time.  This field is used to fill the pause\_time field of outgoing flow-control pause frames. The PQT contains 2-octet long unsigned integer indicating the length of time for which the receiving station will be requested to inhibit next data frame transmission The pause\_time is measured in units of pause\_quanta, equal to 512 bit times of the particular implementation (10/100/1000 . Mbit/s)  The range of possible pause\_time is 0 to 65535 pause\_quanta.  Writing a value of 0 disables the flow control protocol. |

FIFO thresholds register CSR19

Address: Operational Base + offset (0x098)

Table 478.FIFO thresholds register CSR19

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | RW | 0x0 | FIFO pause threshold level.  The filed contains unsigned value indicating the minimum number of bytes stored in the receive FIFO when the flow control mechanism starts to transmit pause frames.  The actual width of the field depends of RFIFO\_DEPTH parameter and a maximum value is RFIFO\_DEPTH-1 (the significant bits are from 16 to 15+RFIFO\_DEPTH).  Writing a value less than in a FRTL field or 0 disables the flow control reactions on FIFO overflow. |
| 15:0 | RW | 0x0 | FIFO restart threshold level.  The filed contains unsigned value indicating the maximum number of bytes stored in the receive FIFO below which the flow control mechanism stops to transmit pause frames.  The actual width of the field depends of RFIFO\_DEPTH parameter and a maximum value is RFIFO\_DEPTH -1 (the significant bits are from 0 to RFIFO\_DEPTH-1).  Writing a value larger than in a FPTL field or 0 disables the flow control reactions on frame cash overflow. |

Flow control setup & status CSR20

Address: Operational Base + offset (0x0A0)

Table 479.Flow control setup & status CSR20

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | RW | 0x0 | Flow Control Enable  Writing this bit with 1 enables all full and half-duplex flow control modules. The functionality of full-duplex flow control can be configured with TUE, TPE and RPE bits. The functionality of half-duplex flow control (back pressure) can be configured with BPE bit. |
| 30 | RW | 0x0 | Transmit Un-pause frames Enable  Writing this bit with 1 enables the transmission of pause frames with 0 pause time when the FIFO is almost empty (below the programmable threshold levels). |
| 29 | RW | 1’b1 | Transmit Pause frames Enable  Writing this field with 1 enables the transmission of pause frames with when the FIFO is near overflow (above the programmable threshold levels). |
| 28 | RW | 1’b1 | Receive Pause frames Enable.  Setting this bit with 1 enables the reception of pause frames and delaying the transmissions. |
| 27 | RW | 0x0 | Back pressure ( half-duplex flow control ) enable. |
| 26:2 | - | - | Reserved |
| 1 | R | 0x0 | Pause Request Sent.  When read as “1” indicates that the MAC-1G/MAC has sent a pause frame; otherwise read as “0”. |
| 0 | R | 0x0 | Host Transmission Paused.  When read as “1” indicates that the MAC-1G/MAC is now delaying the transmission of the next frame according to the received pause frame; otherwise read as “0”. |

* 1. Programming Sequence



Figure 32.Transmit programming



Figure 33.Receive programming

1. SDIO Controller
   1. Register Summary

Table 480.SDIO Controller Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| HRS00 | 0x00 | HRS\_RSTV00 | General information register |
| HRS01 | 0x04 | HRS\_RSTV01 | Debounce settings register |
| HRS02 | 0x08 | HRS\_RSTV02 | Bus settings register |
| HRS04 | 0x10 | 0x00000000 | HWInit SRS16 configuration for slot #0 |
| HRS05 | 0x14 | 0x00000000 | HWInit SRS17 configuration for slot #0 |
| HRS06 | 0x18 | 0x00000000 | HWInit SRS18 configuration for slot #0 |
| HRS08 | 0x20 | 0x00000000 | HWInit SRS16 configuration for slot #1 |
| HRS09 | 0x24 | 0x00000000 | HWInit SRS17 configuration for slot #1 |
| HRS10 | 0x28 | 0x00000000 | HWInit SRS18 configuration for slot #1 |
| HRS12 | 0x30 | 0x00000000 | HWInit SRS16 configuration for slot #2 |
| HRS13 | 0x34 | 0x00000000 | HWInit SRS17 configuration for slot #2 |
| HRS14 | 0x38 | 0x00000000 | HWInit SRS18 configuration for slot #2 |
| HRS20 | 0x50 | HRS\_RSTV20 | CPRM information/settings register |
| HRS21 | 0x54 | HRS\_RSTV21 | CPRM CBC settings register |
| HRS24 | 0x60 | 0x00000000 | HWInit SRS24 preset for slot #0 |
| HRS25 | 0x64 | 0x00000000 | HWInit SRS25 preset for slot #0 |
| HRS26 | 0x68 | 0x00000000 | HWInit SRS26 preset for slot #0 |
| HRS27 | 0x6C | 0x00000000 | HWInit SRS27 preset for slot #0 |
| HRS28 | 0x70 | 0x00000000 | HWInit SRS24 preset for slot #1 |
| HRS29 | 0x74 | 0x00000000 | HWInit SRS25 preset for slot #1 |
| HRS30 | 0x78 | 0x00000000 | HWInit SRS26 preset for slot #1 |
| HRS31 | 0x7C | 0x00000000 | HWInit SRS27 preset for slot #1 |
| HRS32 | 0x80 | 0x00000000 | HWInit SRS24 preset for slot #2 |
| HRS33 | 0x84 | 0x00000000 | HWInit SRS25 preset for slot #2 |
| HRS34 | 0x88 | 0x00000000 | HWInit SRS26 preset for slot #2 |
| HRS35 | 0x8C | 0x00000000 | HWInit SRS27 preset for slot #2 |
| HRS40 | 0xA0 | 0x00000000 | HWInit SRS56 preset for slot #0 |
| HRS41 | 0xA4 | 0x00000000 | HWInit SRS56 preset for slot #1 |
| HRS42 | 0xA8 | 0x00000000 | HWInit SRS56 preset for slot #2 |
| SRS0 | 0x00 | 0x00000000 | System address/Argument #2 |
| SRS1 | 0x04 | 0x00000000 | Block count and size |
| SRS2 | 0x08 | 0x00000000 | Argument #1 |
| SRS3 | 0x0C | 0x00000000 | Transfer mode & command information |
| SRS4 | 0x10 | 0x00000000 | Response #0 |
| SRS5 | 0x14 | 0x00000000 | Response #1 |
| SRS6 | 0x18 | 0x00000000 | Response #2 |
| SRS8 | 0x20 | 0x00000000 | Buffer data port |
| SRS9 | 0x24 | 0x00000000 | Present state |
| SRS10 | 0x28 | 0x00000000 | Host control settings #0 |
| SRS11 | 0x2C | 0x00000000 | Host control settings #1 |
| SRS12 | 0x30 | 0x00000000 | Interrupt status |
| SRS13 | 0x34 | 0x00000000 | Interrupt status enable |
| SRS14 | 0x38 | 0x00000000 | Interrupt signal enable |
| SRS15 | 0x3C | 0x00000000 | Auto CMD12 Error status |
| SRS16 | 0x40 | HWINIT\_SRS16 | Capabilities #0 |
| SRS17 | 0x44 | HWINIT\_SRS17 | Capabilities #1 |
| SRS18 | ` | HWINIT\_SRS18 | Capabilities #2 |
| SRS20 | 0x50 | 0x00000000 | Force event |
| SRS21 | 0x54 | 0x00000000 | ADMA error status |
| SRS22 | 0x58 | 0x00000000 | ADMA system address |
| SRS24 | 0x60 | HWINIT\_SRS24 | Preset Values #0 |
| SRS25 | 0x64 | HWINIT\_SRS25 | Preset Values #1 |
| SRS26 | 0x68 | HWINIT\_SRS26 | Preset Values #2 |
| SRS56 | 0xE0 | HWINIT\_SRS56 | Shared Bus Control |
| CRS63 | 0xFC | HWINIT\_CRS63 | Host version / Interrupt status |

* 1. Register Descriptions

SDIO\_HRS00

Address: Operational Base + offset (0x00)

Table 481.SDIO\_HRS0 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 30 | RW (Read-Write register) | 0x0 | DDR mode on slot #2  Select DAT bus date rate mode for MMC memory device:  1 = Dual Data Rate is enabled  0 = Single Data Rate is enabled  The DDR mode for SD device should be selected through SDIO\_SRS15.UHSMS. |
| 29 | RW | 0x0 | DDR mode on slot #1  Select DAT bus date rate mode for MMC memory device:  1 = Dual Data Rate is enabled  0 = Single Data Rate is enabled  The DDR mode for SD device should be selected through SDIO\_SRS15.UHSMS. |
| 28 | RW | 0x0 | DDR mode on slot #0  Select DAT bus date rate mode for MMC memory device card:  1 = Dual Data Rate is enabled  0 = Single Data Rate is enabled  The DDR mode for SD device should be selected through SDIO\_SRS15.UHSMS. |
| 26 | RW | 0x0 | MMC-8 mode on slot #2  Select DAT bus width mode for MMC memory device:  1 – 8 bit mode;  0 – 1 or 4 bit mode (depends on SDIO\_SRS10.1 – DTW). |
| 25 | RW | 0x0 | MMC-8 mode on slot #1  Select DAT bus width mode for MMC memory device:  1 – 8 bit mode;  0 – 1 or 4 bit mode (depends on SDIO\_SRS10.1 – DTW). |
| 24 | RW | 0x0 | MMC-8 mode on slot #0  Select DAT bus width mode for MMC memory device:  1 – 8 bit mode;  0 – 1 or 4 bit mode (depends on SDIO\_SRS10.1 – DTW). |
| 18 | HwInit  (Hardware Initialized) | HRS\_RSTV00.19 | Slot #2 available  1 = slot is available (implemented)  0 = slot is not available |
| 17 | HwInit | HRS\_RSTV00.19 | Slot #1 available  1 = slot is available (implemented)  0 = slot is not available |
| 16 | HwInit | HRS\_RSTV00.19 | Slot #0 available  1 = slot is available (implemented)  0 = slot is not available |
| 0 | RWAC  (Read-Write, automatic clear register) | 0x0 | Software reset.  When set to 1, the entire SDIO-HOST is reset. The difference between SWR and SDIO\_SRS11.SRFA is that the latter bit is dedicated to the single slot, while the SWR resets all flip-flops in every slot.After completing the reset operation, SWR bit is automatically cleared. It takes some time to complete the reset operation, so the software should always wait until SWR=0, and continue the other operations only when SWR=0. |

SDIO\_HRS01

Address: Operational Base + offset (0x04)

Table 482.SDIO\_HRS01 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 23:0 | RW | DEBOUNCE | Debounce Period  Defines the number of clk clock cycles used for the debouncing circuit which detects card detection events. The debounce period is equal to:  DP \* ,where is the period of clk clock Typically, DP value should be chosen to obtain the period of 20ms. |

SDIO\_HRS02

Address: Operational Base + offset (0x08)

Table 483.SDIO\_HRS02 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 3:0 | RW | 0 | Programmable Burst Length  The maximum number of beats definition within single DMA burst.   |  |  | | --- | --- | | PBL value | Number of Beats | | 0001b | 1 | | 0010b | 2 | | 0011b | 4 | | 0100b | 8 | | 0101b | 16 | | 0110b | 32 | | 0111b | 64 | | 1000b | 128 | | 1001b | 256 | | 1010b | 512 | | 1011b | 1024 | | others | 2048 | |

SDIO\_HRS4

Address: Operational Base + offset (0x10)

Table 484.SDIO\_HRS04 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo  (Hardware Initialized Write only) | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS05

Address: Operational Base + offset (0x14)

Table 485.SDIO\_HRS05 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS06

Address: Operational Base + offset (0x18)

Table 486.SDIO\_HRS06 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS08

Address: Operational Base + offset (0x20)

Table 487.SDIO\_HRS08 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS9

Address: Operational Base + offset (0x24)

Table 488.SDIO\_HRS09 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS10

Address: Operational Base + offset (0x28)

Table 489.SDIO\_HRS10 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS12

Address: Operational Base + offset (0x30)

Table 490.SDIO\_HRS12 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS13

Address: Operational Base + offset (0x34)

Table 491.SDIO\_HRS13 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS14

Address: Operational Base + offset (0x38)

Table 492.SDIO\_HRS14 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.Read operations on the SDIO\_HRS registers are ignored. |

SDIO\_HRS20

Address: Operational Base + offset (0x50)

Table 493.SDIO\_HRS20 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | ROC  (Read-Only status) | 0 | CPRM busy  1 = CPRM is busy.  0 = CPRM is ready to execute next command. |
| 30\* | RO  (Read-Only register) | 0 | CPRM clock stable  1 = clock is stable  0 = clock is unstable  The CPRM block should be used only if this bit is equal to 1 |
| 29\* | RW | 0 | CPRM clock enable  This register can be used to gating clock to CPRM component.  1 = clock is enabled  0 = clock is disabled |
| 23:16 | ROC | 0 | Copy Control Information Output  The CCI calculated during the Title Key decryption. |
| 15:8 | RW | HRS\_RSTV20.(15..8) | Copy Control Information Input  The CCI that is to be encrypted together with a random Title Key. |
| 7 | RW | HRS\_RSTV20.7 | CPRM enable  1 = enables CPRM operation (all card read/write operations are affected).  0 = CPRM does not affect any  operation of SDIO-HOST. |
| 6 | RW | HRS\_RSTV20.6 | CPRM halt  1 = halts the operation of CRPM module (only in case of MKB processing and CBC enabled transfers).  0 = normal CPRM operation. |
| 3:0 | RW | HRS\_RSTV20.(3..0) | CPRM command  The index of CPRM command.   |  |  | | --- | --- | | CRPM\_CMD value | Command mnemonic | | 1h | PROCESS\_MKB | | 2h | PROCESS\_MKB\_EXT | | 3h | CALC\_KMU | | 4h | AKE | | 5h | SET\_BIND\_ID | | 8h | GEN\_KT | | 9h | CALC\_KT | | Ah | GEN\_KT\_BIND | | Bh | CALC\_KT\_BIND | | Ch | DECRYPT\_USER | | Dh | ENCRYPT\_USER | | Eh | DECRYPT\_PROT | | Fh | ENCRYPT\_PROT | | Others | NOP | |

SDIO\_HRS21

Address: Operational Base + offset (0x54)

Table 494.SDIO\_HRS Register Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:21 | - | - | Reserved |
| 20:16 | RW | HRS\_RSTV21.(20..16)  (11111b) | CBC block size  If SD\_BIND\_BLOCK = 11111b the whole file is considered as a single CBC block; otherwise the file is encrypted/decrypted in CBC mode with a block size equal to 64\*bytes. Each block starts a new CBC chain. |
| 15:4 | - | - | Reserved |
| 3:0 | RW | HRS\_RSTV21.(3..0)  (0000b) | Unencrypted header size  If SD\_BIND\_HEADER = 0000b, no header; otherwise the device passes first 64\* bytes (header) without  processing. |

SDIO\_HRS24

Address: Operational Base + offset (0x60)

Table 495.SDIO\_HRS24 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.  This register are read 0 |

SDIO\_HRS25

Address: Operational Base + offset (0x64)

Table 496.SDIO\_HRS25 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.This register are read 0 |

SDIO\_HRS26

Address: Operational Base + offset (0x68)

Table 497.SDIO\_HRS26 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS27

Address: Operational Base + offset (0x6C)

Table 498.SDIO\_HRS27 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS28

Address: Operational Base + offset (0x70)

Table 499.SDIO\_HRS28 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS29

Address: Operational Base + offset (0x74)

Table 500.SDIO\_HRS29 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS30

Address: Operational Base + offset (0x78)

Table 501.SDIO\_HRS30 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS31

Address: Operational Base + offset (0x7C)

Table 502.SDIO\_HRS31 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS32

Address: Operational Base + offset (0x80)

Table 503.SDIO\_HRS32 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.This register are read 0 |

SDIO\_HRS33

Address: Operational Base + offset (0x84)

Table 504.SDIO\_HRS33 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.This register are read 0 |

SDIO\_HRS34

Address: Operational Base + offset (0x88)

Table 505.SDIO\_HRS34 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value. This register are read 0 |

SDIO\_HRS35

Address: Operational Base + offset (0x8C)

Table 506.SDIO\_HRS35 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | HwInit Wo | 0 | The value written to the SDIO\_HRS field updates the corresponding SDIO\_SRS value.This register are read 0 |

SDIO\_SRS0

Address: Operational Base + offset (0x00)

Table 507.SDIO\_SRS0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0 | **SDMA System Address/Argument #2 Register**  This field is used as:   * System Address Register when Simple DMA   engine is enabled (SDIO\_SRS3.DMAE=1 and  SDIO\_SRS10.DMASEL=0) for the data transfer.   * Argument for CMD23 when CPU   (SDIO\_SRS3.DMAE=0) or ADMA2 engine  (SDIO\_SRS3.DMAE=1 and SDIO\_SRS10.DMASEL=2) is  enabled and ACMD23 is selected (SDIO\_SRS.ACMD=2).  **Simple DMA System Address**  This field is providing start address for the SDMA engine, and is resuming DMA transfer after DMA interrupt.The software should not read or write SDMA SA when the SDMA transaction is in progress. After SDMA is stopped (finishing data transfer command or DMA interrupt), SDMA SA points to the next system address of the next contiguous data position.  **Argument #2**  This register contains value that is used as an argument for the Auto CMD23. The value indicates the number of blocks that will be transferred during multi-block transaction.This maximum accepted value for this field is FFFFFFFFh (for ADMA2 transfer), and FFFFh (for CPU).When the Argument #2 is set 0 no data blocks will  be transferred. |

SDIO\_SRS1

Address: Operational Base + offset (0x04)

Table 508.SDIO\_SRS1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:16 | RW,  P(write protected) | 0 | **Blocks Count For Current Transfer**  Contains the number of data blocks to be transferred.  If Block Count is enabled (SDIO\_SRS3.BCE = 1) the value is decremented after each block transfer.  This value is ignored when SDIO\_SRS3.BCE = 0.When BCCT is 0, no data blocks will be transferred. This register should be accessed only when no data transfer is in progress. During the data transfer, read operations on this register may return an invalid value, and write operations are ignored. This register is enabled when Block Count Enable in the  SDIO\_SRS3 register is set to 1 and is valid only for multiple block transfers   |  |  | | --- | --- | | BCCT value | Number of blocks | | 0000h | 0 | | 0001h | 1 | | … | … | | FFFFh | 65535 | |
| 14:12 | RW | 0 | **Host DMA Buffer Boundary**  Contains the size of the contiguous buffer in the  system memory. When the SDMA reaches the buffer boundary, it stops the SDMA transaction and  generates the DMA Interrupt (SDIO\_SRS12.DMAINT).  After the DMA Interrupt, the software should write  new SDMA System Address (SDIO\_SRS0) or set Continue  Request (SDIO\_SRS10.CR) in order to resume the SDMA  transaction. If Continue Request is used without  changing SDIO\_SRS0, then the SDMA starts at the next  address after the buffer boundary.   |  |  | | --- | --- | | HDMABB | buffer boundary | | 000b | 4KB | | 001b | 8KB | | 010b | 16KB | | 011b | 32KB | | 100b | 64KB | | 101b | 128KB | | 110b | 256KB | | 111 | 512KB | |
| 11:0 | RW，P | 0 | **Transfer Block Size**  Specifies the block size for block data transfers.  Values ranging from 1 up to the maximum buffer size can be set. TBS can be accessed only if no data transfer is in progress. Read operations during the data transfer may return an invalid value, and write operations are ignored.  TBS should never exceed the FIFO buffer size physically implemented inside the SDIO-HOST. The FIFO buffer size is equal to , where FIFODEPTH is the generic parameter of the core.The memory uses block size up to 512 bytes.   |  |  | | --- | --- | | TBS | Block Size | | 0800h | 2048 Bytes | | 0799h | 2047 Bytes | | … | … | | 200h | 512 Bytes | | … | … | |

SDIO\_SRS2

Address: Operational Base + offset (0x08)

Table 509.SDIO\_SRS2 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0 | Command argument  Contains bits 39.8 of command argument. |

SDIO\_SRS3

Address: Operational Base + offset (0x0c)

Table 510.SDIO\_SRS3 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 29:24 | RW | 0 | Command Index  Contains an index of the command to be sent. The  index should be in range 0-63, corresponding directly to CMD0-CMD63 (or ACMD0-ACMD63) commands as  shown below:   |  |  | | --- | --- | | CI value | Command Mnemonic | | 000000b | CMD0 / ACMD0 | | 000001b | CMD1 / ACMD1 | | … | … | | 111111b | CMD63 / ACMD63 |   Writing the Command Index triggers the actual command generation. This field should be written only when Command Inhibit CMD bit is 0 in Present  State Register (SDIO\_SRS9).  To check the list of available commands, please refer  to the appropriate Card/Device specifications (SD,  SDIO, or MMC Specifications). |
| 23:22 | RW | 0 | **Command Type**  This field defines the command type. There are 4  command types:   * Abort Command (11b) used when the software wants to stop the current data transfer (read or write data transfer). After sending an Abort Command, the software will also issue the software reset. * Suspend Command (01b) used by the   software to release the SD bus.   * Resume Command (10b) used to restart the   transaction previously suspended by the Suspend Command.   * Normal Command (00b) covers all the remaining commands. |
| 21 | RW | 0 | Data Present Select  Should be set to 1 for commands which transfer data  (i.e. read or write data from the card using DAT line).Should be 0 for all other commands, including:   * Commands using only CMD line * Commands with busy (SDIO\_SRS3.RTS=11b) * Resume type Commands (SDIO\_SRS3.CT=10b) |
| 20 | RW | 0 | Command Index Check Enable  When set to 1, the SDIO-HOST checks if the Command index field of the response is equal to the  SDIO\_SRS3.CI value.  When 0, the check is not performed and Command  index field of the response is ignored. |
| 19 | RW | 0 | Command CRC(Cyclic Redundancy Check) Check Enable  When set to 1, the SDIO-HOST checks if the CRC  field of the response is valid.  When 0, the CRC check is disabled and the CRC fieldof the response is ignored.  The CRC check should be disabled for responses  which do not contain an actual CRC value (some  responses contain all 1's in place of the CRC field),  and enabled for all other kinds of responses. |
| 17:16 | RW | 0 | Response Type Select  Defines the expected response length as below:   |  |  | | --- | --- | | RTS value | Response Type | | 00b | no response | | 01b | 136-bit response | | 10b | 48-bit response | | 11b | 48-bit response with BUSY |   Every command implies one of the response types listed above. To check the response type corresponding to a given command, please refer to  the appropriate card/device specifications(SD,SDIO, or MMC Specifications). |
| 15:6 | - | - | Reserved |
| 5 | RW, P | 0 | Multi/Single Block Select  When set to 1, indicates multi block data transfer.  When 0, indicates single block transfer.  This field is hardware-protected by Command Inhibit DAT bit in Present State Register (SDIO\_SRS9). When SDIO\_SRS9.CIDAT=1, all writes to this field are ignored. |
| 4 | RW, P | 0 | Data Transfer Direction Select  Selects between read and write transactions for commands which transfer data (i.e. with SDIO\_SRS3.DPS=1).  Should be set to 1 for data read operations (data  read from the card to the host).  Should be 0 for data write operations (data write from the host to the card).  For commands which do not transfer data (SDIO\_SRS3.DPS=0), DTDS is treated as a don't care by  the SDIO-HOST.  This field is hardware-protected by Command Inhibit DAT bit in Present State Register (SDIO\_SRS9). When SDIO\_SRS9.CIDAT=1, all writes to this field are ignored. |
| 3:2 | RW, P | 0 | Auto CMD Enable  The field is used to send one additional command to  the device when the command is issued.   |  |  | | --- | --- | | Value | Command | | 00b | No auto command | | 01b | Auto CMD12 | | 10b | Auto CMD23 | | 11b | Reserved |   Auto CMD12 (01b) The Host sends CMD12(Abort) automatically when last block of multiblock transfer is completed.  Auto CMD23 (10b) The Host sends CMD23 (Set Block Count) automatically before issued transfer data command. The argument of this command is to be set in SDIO\_SRS0 register. Because the SDIO\_SRS0 is used to store the argument ,this command shall not be used when SDMA engine is selected.  Auto CMD disable (00b) If auto command is not required this field should be set 00b. The reserved value is prohibited and shall not be used.  This field is hardware-protected by Command Inhibit DAT bit in Present State Register (SDIO\_SRS9). When SDIO\_SRS9.CIDAT=1, all writes to this field are ignored. |
| 1 | RW, P | 0 | Block Count Enable  When set to 1, the SFR1.BCCT Block Count is automatically decremented after each data block being transfer between host and card/device on DAT  Line .When 0, block counting is disabled, and SFR1.BCCT retains its value. This is useful in executing an infinite transfer.  In case of using CPU or SDMA modes, the software  would use an explicit ABORT type command to stop  the infinite transaction.  In case of ADMA2 mode, the total data transfer  length is designated by the descriptors.  This field is hardware-protected by Command Inhibit DAT bit in Present State Register (SDIO\_SRS9). When SDIO\_SRS9.CIDAT=1, all writes to this field are ignored. |
| 0 | RW, P | 0 | DMA Enable  When set to 1, it enables DMA functionality. DMA can be enabled only if it is supported as indicated in the DMA Support in the SFR16 register. If DMA is not supported, this bit is treated as a don't care bit .This field is hardware-protected by Command Inhibit DAT bit in Present State Register (SDIO\_SRS9). When SDIO\_SRS9.CIDAT=1, all writes to this field are ignored. |

SDIO\_SRS4

Address: Operational Base + offset (0x10)

Table 511.SDIO\_SRS4 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | ROC | 0 | Bits: 31..0 of command response (RESP). |

SDIO\_SRS5

Address: Operational Base + offset (0x14)

Table 512.SDIO\_SRS5 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | ROC | 0 | Bits: 63..32 of command response (RESP). |

SDIO\_SRS6

Address: Operational Base + offset (0x18)

Table 513.SDIO\_SRS6 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | ROC | 0 | Bits: 95..64 of command response (RESP). |

SDIO\_SRS7

Address: Operational Base + offset (0x1C)

Table 514.SDIO\_SRS7 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | ROC | 0 | Bits: 127..96 of command response (RESP). |

SDIO\_SRS8

Address: Operational Base + offset (0x20)

Table 515.SDIO\_SRS8 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0 | **Buffer Data Port**  The field is to access the internal buffer (data block) in CPU transfer mode. 8-bit, 16-bit, or 32-bit access to SDIO\_SRS8 is possible with the following restrictions:   * Only sequential contiguous access in Little Endian mode is possible. For example, if the software accesses SDIO\_SRS8.7..0, then the next transfer will access SDIO\_SRS8.15..8. No byte skipping is allowed. * Each new block will start at the least significant byte of SDIO\_SRS8, which is SDIO\_SRS8.7...0. * If the block size is not a multiple of 32-bits, and the software accesses SDIO\_SRS8 using 32-bit words, then the excess bytes of the last word are ignored. This allows the software driver to use only 32-bit data transfers regardless of the block size. * Access to the register with precaution – the FIFO pointers can be damaged when buffer is not ready or when number of accesses exceed the transfer block size (SDIO\_SRS02.11..0/TBS).   Below is the table with all transfers (Byte enable patterns)  which are allowed on SDIO\_SRS8:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | transfer  width | be(3) | be(2) | be(1) | be(0) | | BDP(31:24) | BDP(23:16) | BDP(15:8) | BDP(7:0) | | 32-bit | 1 | 1 | 1 | 1 | | 16-bit | 0 | 0 | 1 | 1 | | 16-bit | 1 | 1 | 0 | 0 | | 8-bit | 0 | 0 | 0 | 1 | | 8-bit | 0 | 0 | 1 | 0 | | 8-bit | 0 | 1 | 0 | 0 | | 8-bit | 1 | 0 | 0 | 0 | |

SDIO\_SRS9

Address: Operational Base + offset (0x24)

Table 516.SDIO\_SRS9 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:25 | Reserved | - | Reserved |
| 24 | RO | 0 | CMD Line Signal Level  The value is equal to the actual signal level on CMD  line of the SD interface (cmd\_s#\_i).  Is useful for debugging purposes. |
| 23:20 | RO | 0 | DAT[3:0] Line Signal Level  The value is equal to the actual signal level on DAT  pins of the SD interface:  SFR9.23 – dat3\_i pin level.  SFR9.22 – dat2\_i pin level.  SFR9.21 – dat1\_i pin level.  SFR9.20 – dat0\_i pin level. |
| 19 | RO | 0 | Write Protect Switch Pin Level  The value is equal to the actual signal level on Write  Protect pin of the SD interface (sdwp\_s#\_n).  1 - means that the write operation is enabled.  0 - means that the write operation is disabled. |
| 18 | RO | 0 | Card Detect Pin Level  The value is equal to the inversed signal level on  Card Detect pin of the SD interface (sdwp\_s#\_n).  1 – means that the card is inserted.  0 – means no card is inside the slot.  Debouncing is not performed on CDL, therefore the  use of Card Inserted (SFR9.CI) bit is recommended  during normal work.  CDL bit is useful only for debugging purposes. |
| 17 | RO | 0 | Card State Stable  Indicates if Card Detect Pin Level (CDL) is stable.  1 – means that the CDL value is stable.  0 – means that the CDL is not stable (during card  insertion/removal or during the reset). Is useful for debugging purposes. |
| 16 | RO | 0 | Card Inserted  Indicates if the card is inserted inside the slot.  1 – means that the card is inserted.  0 – means no card is inside the slot.  Unlike CDL, value of CI bit is guaranteed to be table  (i.e. debouncing is performed on this bit).  Use of this bit is recommended during the normal  operation of SDIO-HOST. |
| 11 | ROC | 0 | Buffer Read Enable  Shows the current data buffer (SDIO\_SRS8) state during  CPU data read transfers.  1 – valid data can be read from the data buffer.  0 – no valid data inside the data buffer.  After reading the entire data block, this changes to 0. |
| 10 | ROC | 0 | Buffer Write Enable  Shows the current data buffer state during CPU data  write transfers.  1 – data can be written to the data buffer.  0 – data cannot be written.  After reading the entire data block, this changes to 0. |
| 9 | ROC | 0 | Read Transfer Active  Indicates the status of the read data transfer.  1 – means that the data read transfer is in progress.  0 – means that no read transfer is in progress.  The SDIO-HOST sets RTA to 1 after sending the read command, or after restarting the read transfer by the Continue Request (SDIO\_SRS10.CR).  This is cleared by the hardware after the last block of  the read transfer, or after stopping the read transfer  by the Stop at Block Gap Request (SDIO\_SRS10.SBGR).  In both cases, the entire data has to be read by the  system from the data buffer before clearing RTA bit.  In other words, RTA=0 means that the entire data is  already transferred to the system, and internal data  buffer is empty.  In the case of ADMA transfers, the end of read transfer is designated by the END flag in a descriptor. |
| 8 | ROC | 0 | Write Transfer Active  Indicates the status of the write data transfer.  1 – means that the data write transfer is in progress.  0 – means that no write transfer is in progress.  The SDIO-HOST sets WTA to 1 after sending the  write command, or after restarting the write transfer  by the Continue Request (SDIO\_SRS10.CR).  This is cleared by the hardware after the last block of  the write transfer, or after stopping the write transfer  by the Stop at Block Gap Request (SDIO\_SRS10.SBGR). In both cases, the entire data has to be transferred  to the card from the internal data buffer before  clearing WTA bit. In other words, RTA=0 means that  the entire data is already transferred to the card, and  CRC response for the last data block is already received. In the case of ADMA transfers, the end of write transfer is designated by the END flag in a  descriptor. |
| 3 | ROC | 0 | Re-Tuning Request  The signal will be used by the SD driver to rerun the Sample Clock Tuning operation. This field is available only if the Sample Clock Select（SDIO\_SRS15.SCS) is 1.This field is directly supplied by the external port sdrtr.Changing from 0 to 1, sets SDIO\_SRS12.12 (RTNGE). |
| 2 | ROC | 0 | DAT Line Active.  Indicates if the DAT lines of SD interface are currently in use.  1 – means that DAT lines are active (in use).  0 – means that DAT lines are released (not in use).  This is set when Read or Write Transfer bits are  active (SDIO\_SRS9.RTA=1 or SDIO\_SRS9.WTA=1), or if the card indicates busy state on the DAT lines.The card can become busy immediately after the write operation, or after command which requires "response with busy".Falling edge of this bit (change from 1 to 0) directly triggers Transfer Complete bit (SDIO\_SRS12.TC). |
| 1 | ROC | 0 | Command Inhibit DAT.  Indicates if the SDIO-HOST can issue a command  which uses DAT line. Commands which use DAT line include write and read data commands and  commands with busy response.  1 – command using DAT line cannot be sent.  0 – command using DAT line can be sent.  When CIDAT=1 then the SDIO\_SRS3.(15..0) is write protected. The software can write SDIO\_SRS3.(15..0) only when CIDAT=0.The behavior of this bit is the same as the behavior of SDIO\_SRS9.DATLA bit (i.e. value of DATLA is equal to the value of CIDAT). |
| 0 | ROC | 0 | Command Inhibit CMD.  Indicates if the SDIO-HOST can issue a command.  1 – command cannot be sent.  0 – command can be sent.  If this bit is 0, indicates the CMD line is not in use  and the Host Controller can issue an SD command  using the CMD line.  This bit is set immediately after the SDIO\_SRS3.CI is  written, indicating start of command transmission.  This bit is cleared when the command response is  received. Even if the Command Inhibit DAT is set to  1, commands using only the CMD line can be issued  if the Command Inhibit CMD is 0. |

SDIO\_SRS10

Address: Operational Base + offset (0x28)

Table 517.SDIO\_SRS10 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 26 | RW | 0 | Wakeup Event Enable On SD Card Removal.  When set to 1, enables wakeup event via Card Removal assertion in the SDIO\_SRS12 register. |
| 25 | RW | 0 | Wakeup Event Enable On Card Inserted.  When set to 1, enables wakeup event via Card Insertion assertion in the SDIO\_SRS12 register. |
| 24 | RW | 0 | Wakeup Event Enable On Card Interrupt.  When set to 1, enables wakeup event via Card  Interrupt assertion in the SDIO\_SRS12 register. |
| 19 | RW | 0 | Interrupt At Block Gap.  When set to 1, enables interrupt detection at the block gap for a multiple block transfer. This bit is valid only in SD4 mode. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. |
| 18 | RW | 0 | Read Wait Control.  When set to 1, enables read wait control. The read  wait function is optional for SDIO cards. If the card  does not support read wait, this bit would never be set to 1; otherwise, DAT line conflict may occur. |
| 17 | RWAC | 0 | Continue Request.  When set to 1, restarts the transfer previously  stopped using the Stop At Block Gap.  The software will clear SDIO\_SRS10.SBGR (Stop At Block Gap) bit before setting the Continue Request.  When SDIO\_SRS10.SBGR=1, then all write operations to  Continue Request are ignored. Clearing SDIO\_SRS10.SBGR can be done before or simultaneously  with writing the Continue Request.  Continue Request bit is cleared automatically by  the SDIO-HOST when SDIO\_SRS9.DATLA (Dat Line Active) changes from 0 to 1, indicating the actual  restart of the transfer. |
| 16 | RW | 0 | Stop At Block Gap Request.  When set to 1, orders the stop executing read and  write transaction at the next block gap for CPU,  SDMA and ADMA transfers. The software will  maintain SBGR=1 until the current transfer is  complete (typically by waiting for SDIO\_SRS12.TC -Transfer Complete bit). After Transfer Complete  event, the software will clear SBGR back to 0.  In the case of the read transfer, the SDIO-HOST  stops after the next data block received from the  card. This uses the ReadWait mechanism if it is  enabled by SDIO\_SRS10.RWC, or stops the card clock  (sdclk\_s#) if ReadWait is disabled.  In the case of the write transfer, SDIO-HOST stops  after the last block written to the data buffer. The  SDIO-HOST sends all data already written to the  data buffer before stopping the transfer. |
| 11:9 | RW | 0 | SD Bus Voltage Select.  Contains a voltage level selection for the SD card.   |  |  | | --- | --- | | BVS value | Voltage level | | 111b | 3.3V (typical) | | 110b | 3.0V (typical) | | 101b | 1.8V (typical) | | others | reserved |   State of BVS directly drives bus\_volt\_s# pins of the  SDIO-HOST. |
| 8 | RW | 0 | SD Bus Power.  When set to 1, the SD device is powered. The state  of this bit directly drives bus\_pow\_s# pin of the SDIO-HOST. Additionally, when BP=0, SDIO-HOST  stops driving CMD, DAT, and SDCLK lines by setting:  cmd\_s#\_en = 0  dat\_s#\_en = 0  and sdclk\_s# = 0  The SDIO-HOST clears BP to 0 automatically when  card is removed from the slot (i.e. after high to low  transition on sdcd\_s#\_n pin). This is to provide the  "hot removal support". |
| 7 | RW | 0 | Card Detect Signal Selection  This bit selects active signal source for card  detection bits and procedures.   |  |  | | --- | --- | | value | Active source | | 1 | CDTL(SDIO\_SRS10.6) bit (for test purposes) | | 0 | sdcd\_s#\_n pin (for normal operation) | |
| 6 | RW | 0 | Card Detect Test Level  This bit can be used for test purposes. When CDSS  bit is set to 1, the value of this bit is used instead  of the input signal sdcd\_s#\_n.   |  |  | | --- | --- | | 1 | card inserted | | 0 | no card |   Please note that the active state of this bit is the  opposite of the active state of the input signal. |
| 5 | RW | 0 | Extended Data Transfer Width  This register controls the DAT Bus Width. If it is 1  then the 8-bit mode is selected. If not, the SDIO\_SRS10.1  (DTW) controls bus width.  This field is valid only when shared bus is not  supported. |
| 4:3 | RW | 0 | DMA Select.  Selects the DMA operating mode. Use of selected  DMA is determined by DMA Enable bit in SDIO\_SRS2  register.   |  |  | | --- | --- | | DMASEL Value | Selected DMA Mode | | 00b | SDMA | | 01b | ADMA1 | | 10b | ADMA2(32-bit Address) | | 11 | Reserved | |
| 2 | RW | 0 | High Speed Enable  Selects operating mode.  High Speed (1) mode selected. The SDIO-HOST  (chip level) drives bidirectional CMD and DAT lines  on the rising edge of SDCLK clock.  Default Speed (0) mode selected. The SDIO-HOST  (chip level) drives bidirectional CMD and DAT lines  on the falling edge of SDCLK clock.  The maximum SD clock frequency is defined as  0-25MHz in the default speed mode, and 0-50MHz  in the High Speed mode. |
| 1 | RW | 0 | Data Transfer Width  SD4 (1) 4-bit mode is selected.  SD1 (0) 1-bit mode is selected.  This bit is ignored when the 8-bit mode is set. |
| 0 | RW | 0 | LED Control  State of this bit directly drives led pin of the SDIOHOST in order to control the external LED diode. LEDC=1 will switch LED on, while LEDC=0 will switch it off.The software will switch LED on to caution the user not to remove the card while the transfer is in progress. |

SDIO\_SRS11

Address: Operational Base + offset (0x2C)

Table 518.SDIO\_SRS11 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:27 | - | - | Reserved |
| 26 | RWAC | 0 | Software Reset For DAT Line  When set to 1, resets the logic related to the data  path, including data buffers and the DMA logic.  The following registers and bits are cleared:  SFR8 register:   * Buffer is cleared.   SFR9 register:   * + Buffer Read Enable   + Buffer Write Enable   + Read Transfer Active   + Write Transfer Active   + DAT Line Active   + Command Inhibit DAT   SFR10 register:   * + Continue Request   + Stop At Block Gap Request   SFR12 register:   * + Buffer Read Ready   + Buffer Write Ready   + DMA Interrupt   + Block Gap Event   + Transfer Complete   After completing the reset operation, SRDAT bit is  automatically cleared. It takes some time to  complete the reset operation, so the software will  wait until SRDAT=0, and continue the other  operations only when SRDAT=0. |
| 25 | RWAC | 0 | Software Reset For CMD Line  When set to 1, resets the logic related to the command/response generation and checking.  The following registers and bits are cleared:  SDIO\_SRS9 register:   Command Inhibit CMD  SDIO\_SRS12 register:   Command Complete  After completing the reset operation, SRCMD bit is  automatically cleared. It takes some time to complete the reset operation, so the software will  wait until SRCMD=0, and continue the other operations only when SRCMD=0. |
| 24 | RWAC | 0 | Software Reset For All  When set to 1, the entire SDIO-HOST is reset.  After completing the reset operation, SRFA bit is  automatically cleared. It takes some time to  complete the reset operation, so the software will  wait until SRFA=0, and continue the other  operations only when SRFA=0.  Additionally, after performing Software reset For  All, the software will reset and reinitialize all cards  inserted to the host. |
| 19:16 | RW | 0 | Data Timeout Counter Value  This value determines the interval by which DAT  line timeouts are detected.  This interval can be computed as below:   |  |  | | --- | --- | | DTCV Value | Timeout Interval | | 1111b | reserved | | 1110b | ﹡ | | 1101b | ﹡ | | … |  | | 0001b | ﹡ | | 0000b | ﹡ |   Where is the sdmclk clock period .Refer to the Data Timeout Error in the SFR12 register for information on factors which generate data timeouts. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the SFR12 register). |
| 15:8 | RW | 0 | SDCLK Frequency Select (lower)  This register is used to select the SDCLK  frequency, i.e. frequency of the card clock  (sdclk\_s# pin).  This value has two different meanings depending  on the Clock Generator Select (SDIO\_SRS11.CGS) bit.  When the SDIO\_SRS11.CGS bit is cleared then the sdclk  frequency is calculated with following expressions:  sdclk = sdmclk; when (N=0)  sdclk= sdmclk/2N; when (N>0)  When the SDIO\_SRS11.CGS bit is set, the Programmable  Clock mode is used. The following expression  helps to calculate clock frequency in this case:  sdlck = (M \* sdmclk)/(N+1), where M is a value  set in SDIO\_SRS17. CLKMPR field.  The value of SDCLKFS register can be changed  only when SDIO\_SRS11.SDCE (SD Clock Enable) = 0. |
| 7:6 | RW | 0 | SDCLK Frequency Select (upper)  This register is an extension for SDIO\_SRS11.15..8  (SDCLKFS). It contains two upper bits of SDCLK  Frequency Select. |
| 5 | RW | 0 | Clock Generator Select  The SDCLK Frequency Divider method can be  selected by this field:  1 - Programmable Clock Mode;  0 - 10-bit Divider Clock Mode.  This bit is fixed to 0 and RO when SDIO\_SRS17.23..16 (CLKMPR) is 0. |
| 2 | RW | 0 | SD Clock Enable  When set to 1, sdclk\_s# clock is enabled.  When cleared to 0, sdclk\_s# clock is stopped.  The SDIO-HOST clears SDCE automatically when  card is removed from the slot (i.e. after the high  to low transition on sdcd\_s#\_n pin). Also, the  sdclk\_s# clock will be stopped by the software  when changing the clock divider (i.e. SDCE bit will  be cleared before writing SDIO\_SRS11.SDCLKFS). |
| 1 | ROC | 0 | Internal Clock Stable.  When read as 1, indicates that the clock on  sdmclk pin of the SDIO-HOST is stable after  setting SDIO\_SRS11.ICE to 1. When read as 0, indicates  that the clock is not stable yet (for example, the  external PLL that generates the clock is not yet  locked).  The value of ICS is equal to the actual signal level  on ics pin of the SDIO-HOST. The user would  connect ics to the external PLL if required.  Otherwise, ics would be connected directly to the  ice output of the SDIO-HOST. |
| 0 | RW | 0 | Internal Clock Enable.  State of this bit controls ice pin of the SDIO-HOST  in order to control the external clock generator(for example, PLL). The ICE bits of every slot are logically OR-ed together and then drive the ice pin  of SDIO-HOST. This means that the ice pin = 0 only when ICE bits = 0 for every slot implemented inside SDIO-HOST. The ice pin = 1 if at least one of the ICE bits is set to 1.When set to 0, the clock on sdmclk pin can be stopped externally. If the sdmclk is stopped, then SDIO-HOST goes to a very low power state. SDIO-HOST registers are still able to be read and written, even if the clock on sdmclk is stopped. Setting of the ICE bit does not affect card  detection. This means that the card detection works even if the clock on sdmclk is stopped. |

SDIO\_SRS12

Address: Operational Base + offset (0x30)

Table 519.SDIO\_SRS12 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 26 | RW1C  (Read-Only status) | 0 | Tuning Error  This bit is set when error is detected in tuning  block. |
| 25 | RW1C | 0 | ADMA Error  Generated when the error occurs during ADMA  read or write transfer.  To resolve the cause of the error, the state of the  ADMA engine at error occurrence is saved in  ADMA Error Status register, and the address of  the descriptor processed at error occurrence is  provided in ADMA System Address register. |
| 24 | RW1C | 0 | Auto CMD Error  Generated when the error occurs during Auto  CMD12/Auto CMD23 command transmission.  It indicates one of the following situations:  1. Detecting that one of the bits in SDIO\_SRS15 register  has changed from 0 to 1.  2. Auto CMD12 is not executed due to the  previous command error. |
| 23 | RW1C | 0 | Current Limit Error  Generated when SDIO-HOST is not supplying  power to SD card due to some failure.  The value is equal to the actual signal level on cle  pin of the SDIO-HOST. |
| 22 | RW1C | 0 | Data End Bit Error  When set to 1, indicates detecting 0 at the end bit  position of read data which uses the DAT line, or  at the end bit position of the Write CRC Status. |
| 21 | RW1C | 0 | Data CRC Error  When set to 1, indicates detecting CRC error when  transferring read data which uses the DAT line, or  when detecting the Write CRC status having a  value of other than "010". |
| 20 | RW1C | 0 | Data Timeout Error  When set to 1, indicates detecting one of the  following timeout conditions:  1. Busy timeout for the response with busy.  2. Busy timeout after Write CRC status.  3. Write CRC Status timeout.  4. Read data timeout. |
| 19 | RW1C | 0 | Command Index Error  When set to 1, indicates that Index error occurs in  the command response. |
| 18 | RW1C | 0 | Command End Bit Error  When set to 1, indicates detecting that the end bit  of a command response is 0. |
| 17 | RW1C | 0 | Command CRC Error  When set to 1, indicates that command CRC error  has occurred. |
| 16 | RW1C | 0 | Command Timeout Error  When set to 1, indicates that no response was  returned within 64 SDCLK cycles from the end bit  of the command. |
| 15 | ROC | 0 | Error Interrupt  If any of the bits in range SDIO\_SRS12(31..16) is set,  this bit is also set. The software can check for an  error by reading this single bit first. |
| 12 | ROC | 0 | Re-Tuning Event  When set to 1, indicates that value on SDIO\_SRS9.3  (RTNGR) is changed from o to 1. |
| 11 | ROC | 0 | Interrupt on line C  This bit is enabled when INT\_C input is asserted  to 0. This bit cannot be cleared. |
| 10 | ROC | 0 | Interrupt on line B  This bit is enabled when INT\_B input is asserted  to 0. This bit cannot be cleared. |
| 9 | ROC | 0 | Interrupt on line A  This bit is enabled when INT\_A input is asserted  to 0. This bit cannot be cleared. |
| 8 | ROC | 0 | Card Interrupt  Indicates the card interrupt. CINT is not sampled  by the card clock, so the interrupt can be detected  even with SD clock stopped (SDIO\_SRS11.SDCE = 0).  Also, CINT is not cleared by writing 1. Instead, the  software will clear the source of an interrupt inside  the card.  After detecting the Card Interrupt, the software  will stop further interrupt detection by clearing  SDIO\_SRS13.CINTE to 0. Then, the software will clear  the interrupt source inside the card by using the  appropriate commands. For the details, please  refer to the given SDIO Card Specification. After  clearing the interrupt source, the card will stop to  drive the interrupt signal to the SDIO-HOST.  Finally, when the interrupt service routine is  finished, the interrupt detection can be enabled by  setting SDIO\_SRS13.CINT back to 1. |
| 7 | RW1C | 0 | Card Removal  Generated when the SDIO\_SRS9.CI bit changes from 1 to 0, indicating card removal.  When read as 1, indicates that the card was removed from the slot.  When read as 0, indicates that the card state is  stable (still inserted or removed) or that the  debouncing is in progress. |
| 6 | RW1C | 0 | Card Insertion  Generated when the SDIO\_SRS9.CI bit changes from 0  to 1, indicating card insertion.  When read as 1, indicates that the card was  inserted to the slot.  When read as 0, indicates that the card state is  stable (still inserted or removed) or that the  debouncing is in progress. |
| 5 | RW1C | 0 | Buffer Read Ready  Generated when the SDIO\_SRS9.BRE changes from 0 to  1, indicating that the data buffer can be read by  the software. |
| 4 | RW1C | 0 | Buffer Write Ready  Generated when the SDIO\_SRS9.BWE changes from 0 to  1, indicating that the data buffer can be written by the software. |
| 3 | RW1C | 0 | DMA Interrupt.  In SDMA mode, DMA interrupt is generated when the Host Controller detects the Host SDMA Buffer boundary.  In ADMA mode, DMA interrupt is generated when the INT flag is set in a currently serviced ADMA descriptor.  Note: Other DMA interrupt factors could be added  in future releases of the SDIO-HOST. |
| 2 | RW1C | 0 | Block Gap Event  Generated when the read/write transaction is  stopped at a block gap as the result of setting  SDIO\_SRS10.SBGR to 1. |
| 1 | RW1C | 0 | Transfer Complete  Generated when the transfer which uses the DAT line is complete. Transfers which use the DAT line include the read/write transfers and commands with a busy response.  In the case of the read transfer, TC indicates that the entire data was transferred from the card to the host system (i.e. the SDIO-HOST FIFO is empty after reading the last data block).In the case of the write transfer, TC indicates that the entire data was transferred from the SDIO-HOST to the card (i.e. the SDIO-HOST FIFO is empty after writing the last data block), and the card accepted the data (busy signal released after the last block).  In the case of the command with a busy response,  TC indicates that the busy signal is released after the response. |
| 0 | RW1C | 0 | Command Complete  Generated when the end bit of the response is received, except the response for Auto-CMD12 command. Auto-CMD12 command does not  generate CC. |

SDIO\_SRS13

Address: Operational Base + offset (0x34)

Table 520.SDIO\_SRS13 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 26 | RW | 0 | Tuning Error status enable  1 – enabled  0 – masked |
| 25 | RW | 0 | ADMA Error status enable  1 – enabled  0 – masked |
| 24 | RW | 0 | Auto CMD12 Error status enable  1 – enabled  0 – masked |
| 23 | RW | 0 | Current Limit Error status enable  1 – enabled  0 – masked |
| 22 | RW | 0 | Data End Bit Error status enable  1 – enabled  0 – masked |
| 21 | RW | 0 | Data CRC Error status enable  1 – enabled  0 – masked |
| 20 | RW | 0 | Data Timeout Error status enable  1 – enabled  0 – masked |
| 19 | RW | 0 | Command Index Error status enable  1 – enabled  0 – masked |
| 18 | RW | 0 | Command End Bit Error status enable  1 – enabled  0 – masked |
| 17 | RW | 0 | Command CRC Error status enable  1 – enabled  0 – masked |
| 16 | RW | 0 | Command Timeout Error status enable  1 – enabled  0 – masked |
| 15 | RO | 0 | Fixed to 0 |
| 12 | RW | 0 | Re-Tuning Event status enable  1 – enabled  0 – masked |
| 11 | RW | 0 | Interrupt on line C status enable  1 – enabled  0 – masked |
| 10 | RW | 0 | Interrupt on line B status enable  1 – enabled  0 – masked |
| 9 | RW | 0 | Interrupt on line A status enable  1 – enabled  0 – masked |
| 8 | RW | 0 | Card Interrupt status enable  1 – enabled  0 – masked |
| 7 | RW | 0 | Card Removal status enable  1 – enabled  0 – masked |
| 6 | RW | 0 | Card Insertion status enable  1 – enabled  0 – masked |
| 5 | RW | 0 | Buffer Read Ready status enable  1 – enabled  0 – masked |
| 4 | RW | 0 | Buffer Write Ready status enable  1 – enabled  0 – masked |
| 3 | RW | 0 | DMA Interrupt status enable  1 – enabled  0 – masked |
| 2 | RW | 0 | Block Gap Event status enable  1 – enabled  0 – masked |
| 1 | RW | 0 | Transfer Complete status enable  1 – enabled  0 – masked |
| 0 | RW | 0 | Command Complete status enable  1 – enabled  0 – masked |

SDIO\_SRS14

Address: Operational Base + offset (0x38)

Table 521.SDIO\_SRS14 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 26 | RW | 0 | Tuning Error interrupt enable  1 – enabled  0 – masked |
| 25 | RW | 0 | ADMA Error interrupt enable  1 – enabled  0 – masked |
| 24 | RW | 0 | Auto CMD12 Error interrupt enable  1 – enabled  0 – masked |
| 23 | RW | 0 | Current Limit Error interrupt enable  1 – enabled  0 – masked |
| 22 | RW | 0 | Data End Bit Error interrupt enable  1 – enabled  0 – masked |
| 21 | RW | 0 | Data CRC Error interrupt enable  1 – enabled  0 – masked |
| 20 | RW | 0 | Data Timeout Error interrupt enable  1 – enabled  0 – masked |
| 19 | RW | 0 | Command Index Error interrupt enable  1 – enabled  0 – masked |
| 18 | RW | 0 | Command End Bit Error interrupt enable  1 – enabled  0 – masked |
| 17 | RW | 0 | Command CRC Error interrupt enable  1 – enabled  0 – masked |
| 16 | RW | 0 | Command Timeout Error interrupt enable  1 – enabled  0 – masked |
| 12 | RW | 0 | Re-Tuning Event interrupt enable  1 – enabled  0 – masked |
| 11 | RW | 0 | Interrupt on line C interrupt enable  1 – enabled  0 – masked |
| 10 | RW | 0 | Interrupt on line B interrupt enable  1 – enabled  0 – masked |
| 9 | RW | 0 | Interrupt on line A interrupt enable  1 – enabled  0 – masked |
| 8 | RW | 0 | Card Interrupt interrupt enable  1 – enabled  0 – masked |
| 7 | RW | 0 | Card Removal interrupt enable  1 – enabled  0 – masked |
| 6 | RW | 0 | Card Insertion interrupt enable  1 – enabled  0 – masked |
| 5 | RW | 0 | Buffer Read Ready interrupt enable  1 – enabled  0 – masked |
| 4 | RW | 0 | Buffer Write Ready interrupt enable  1 – enabled  0 – masked |
| 3 | RW | 0 | DMA Interrupt enable  1 – enabled  0 – masked |
| 2 | RW | 0 | Block Gap Event interrupt enable.  1 – enabled  0 – masked |
| 1 | RW | 0 | Transfer Complete interrupt enable.  1 – enabled  0 – masked |
| 0 | RW | 0 | Command Complete interrupt enable.  1 – enabled  0 – masked |

SDIO\_SRS15

Address: Operational Base + offset (0x3C)

Table 522.SDIO\_SRS15 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31 | RW | 0 | Preset Value Enable  If this bit is set, the values from preset registers  SDIO\_SRS24 – SDIO\_SRS27 are used for update SDIO\_SRS11.15..8  (SDCLKFS), SDIO\_SRS11.7..6/USDCLKFS, SDIO\_SRS11.5/CGS,  SDIO\_SRS21.21..20/DSS. The register used to update  process depends on the selected speed mode. |
| 30 | RW | 0 | Asynchronous Interrupt Enable  This bit is used in 4-bit SD mode only. When this  bit is selected the interrupt from SDIO device will  be received by the host even if the SDCLK (SD  interface) clock is disabled. |
| 23 | RW | 0 | Sampling Clock Select  The host updates this bit when the tuning  procedure is finished. If this bit is set to 1, this  means that the tuning procedure is successfully  completed. Otherwise it means that procedure  failed and clock tuning logic is disabled.  This bit is valid only after the procedure is finished  Writing 1 will be ignored.  Writing 0 will reset and disable tuning block |
| 22 | RW | 0 | Execute Tuning  This register controls tuning procedure.  The procedure starts when the bit is set 1.  The procedure can be aborted when the bit is  cleared.  The bit is read 1 while the procedure is in progress, and 0 when the procedure is finished.   |  |  |  | | --- | --- | --- | | SCS | EXTNG | result | | 0 | 0 | Reset and disable clock tuning  logic | | 0 | 1 | Reset and restart tuning process | | 1 | 0 | Stop tuning procedure | | 1 | 1 | Start retuning (without clock tuning logic reset) | |
| 21:20 | RW | 0 | Driver Strength Select  This bit controls the electric parameters of I/O driver. Up to 4 configurations of I/O driver settings can be implemented.   |  |  | | --- | --- | | 00 | Driver Type B (default) | | 01 | Driver Type A | | 10 | Driver Type C | | 11 | Driver Type D |   The bit is irrelevant when the SDIO\_SRS15.19 (1V8SL) is  cleared. |
| 19 | RW | 0 | 1.8V Signaling Enable  This bit is to switch I/O signaling voltage level on the SD interface between 3.3V and 1.8V. When set 1, the 1.8V level is selected. When set 0, the 3.3V is selected. |
| 18:16 | RW | 0 | UHS Mode Select  This register is used to select one of UHS modes:   |  |  | | --- | --- | | Value | UHS-I Mode | | 000b | SDR12 | | 001b | SDR25 | | 010b | SDR50 | | 011b | SDR104 | | 100b | DDR50 | | 101b | Reserved | | 110b | Reserved | | 111b | Reserved |   The mode selected by this field is active only when 1.8V signaling level is active (SDIO\_SRS15.19 (1V8SL) bit is set). |
| 7 | ROC | 0 | Command Not Issued By Auto CMD12 Error When read as 1, the command was not executed by the Host due to the previous Auto CMD12 error.  When Host detects any error during Auto CMD12, then all further command generation attempts are blocked. The software reset sequence is needed for recovery.  Bit is updated with 0, when Auto CMD23 Error is detected (any of bits SDIO\_SRS15.4-SDIO\_SRS15.1 is set). |
| 4 | ROC | 0 | Auto CMD Index Error  When read as 1, means that Command Index error occurred in the Auto CMD response. |
| 3 | ROC | 0 | Auto CMD Index Error  When read as 1, means that Command Index error occurred in the Auto CMD response. |
| 2 | ROC | 0 | Auto CMD CRC Error  When read as 1, indicates a CRC error was detected in the Auto CMD response or conflict on the CMD lines is detected.   |  |  |  | | --- | --- | --- | | SDIO\_SRS15.2  / ACCE | SDIO\_SRS15.1  / ACTE | Status | | 0 | 0 | No error | | 0 | 1 | Auto CMD Timeout error  detected | | 1 | 0 | Auto CMD CRC error  detected | | 1 | 1 | Conflict on the CMD line  detected | |
| 1 | ROC | 0 | Auto CMD Timeout Error  When read as 1, indicates that no response is returned within 64 sdclk\_s# cycles from the end bit of the Auto CMD or conflict on the CMD lines is detected (see table in SDIO\_SRS15.2 field description).If this bit is set to 1, the other error status bits (D04-D0 2 ) are meaningless. |
| 0 | ROC | 0 | Auto CMD12 Not Executed  When set to 1, means the SDIO-HOST cannot issue Auto CMD12 due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.  Bit is updated with 0, when Auto CMD23 Error is  detected (any of bits SDIO\_SRS15.4-SDIO\_SRS15.1 is set). |

SDIO\_SRS16

Address: Operational Base + offset (0x40)

Table 523.SDIO\_SRS16 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:30 | HwInit | HwInit | Slot Type  These bits inform what type of slot is provided.   |  |  | | --- | --- | | 00 | Removable Card Slot | | 01 | Embedded Slot for One Device | | 10 | Shared Bus Slot | | 11 | Reserved | |
| 29 | HwInit | HwInit | Asynchronous Interrupt Support  1 – Asynchronous mode supported  0 – Asynchronous mode not supported |
| 28 | HwInit | HwInit | 64-bit System Bus Support  This bit is always set to 0 to indicate that core  does not support 64-bit system bus. |
| 26 | HwInit | HwInit | Voltage Support 1.8V  1 – 1.8V supported  0 – 1.8V not supported |
| 25 | HwInit | HwInit | Voltage Support 3.0V  1 – 3.0V supported  0 – 3.0V not supported |
| 24 | HwInit | HwInit | Voltage Support 3.3V  1 – 3.3V supported  0 – 3.3V not supported |
| 23 | HwInit | HwInit | Suspend/Resume Support  1 – Suspend/Resume is enabled  0 – Suspend/Resume disabled |
| 22 | HwInit | HwInit | DMA Support  1 – DMA (SDMA mode) supported  0 – DMA (SDMA mode) not supported  This bit defines whether the DMA is supported in the given slot or not. There is a separate  IMPLEMENT\_DMA generic parameter which  defines whether the DMA is physically implemented inside SDIO-HOST. If DMAS is set to 1 for any card slot, then the IMPLEMENT\_DMA would also be set to 1. |
| 21 | HwInit | HwInit | High Speed Support  1 – High Speed mode supported  0 – High Speed mode not supported |
| 20 | HwInit | HwInit | ADMA1 Support  1 – ADMA1 mode supported  0 – ADMA1 mode not supported |
| 19 | HwInit | HwInit | ADMA2 Support  1 – ADMA2 mode supported  0 – ADMA2 mode not supported |
| 18 | HwInit | HwInit | 8-bit Embedded Device Support  1 – 8-bit mode supported  0 – 8-bit mode not supported  If this bit is 0, the SDIO\_SRS10.5 (EDTW) register is not  implemented. |
| 17:16 | HwInit | HwInit | Max Block Length  This value indicates the maximum block size that  can be transferred by the SDIO-HOST. Three sizes  can be defined as indicated below:   |  |  | | --- | --- | | MBL Value | Maximum Block Size | | 00b | 512 Bytes | | 01b | 1024 Bytes | | 10b | 2048 bytes | | 11b | reserved | |
| 15:8 | HwInit | HwInit | Base Clock Frequency For SD Clock  Field defines the base clock frequency for the SD  Clock in 1MHz units. The base clock is the clock  supplied on sdmclk pin of the SDIO-HOST. The maximum clock frequency supported is between  10MHz to 255MHz.  If BCSDCLK = 0, the Host System has to obtain  the clock information via another method (i.e. not  defined by the specification). |
| 7 | HwInit | HwInit | Timeout Clock Unit  Field defines the frequency unit for the  SDIO\_SRS16.TCF.  0 – kHz  1 – MHz |
| 5:0 | HwInit | HwInit | Timeout Clock Frequency  Defines the base clock frequency used to detect  Data Timeout Error. The SDIO\_SRS16.TCU bit  determines the unit used.   |  |  |  | | --- | --- | --- | | TCF value | Timeout Clock Frequency | | | TCF = 0 | TCF = 1 | | 111111b | 63 kHz | 63 MHz | | 111110b | 62 kHz | 62 MHz | | … | … | … | | 000001b | 1kHz | 1MHz | | 000000b | Host System has to obtain the  clock information via another  method (i.e. not defined by the  spec.). | | |

SDIO\_SRS17

Address: Operational Base + offset (0x44)

Table 524.SDIO\_SRS17 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 23:16 | HwInit | HwInit | Clock Multiplier  This filed contains value used to generate sdclk in  programmable clock mode.  This field is to be 0 – the core does not support  clock multiplier. |
| 15:14 | HwInit | HwInit | Re-Tuning Modes  Depending on the retuning method, the some restrictions are assumed for the data length between re-tunings.  The core can work with supporting one of the three method:  0 – Mode1: The software driver will use timer to  calculate when the re-tuning is to be rerun. The data length between operations is limited to the 4MB.  1 – Mode2: The driver will use either the retuning request (external input pin sdrtr is used for this purpose) or timer to predict when next retuning should be performed. The data length between operations is limited to the 4MB.  2 – Mode3: This mode is similar to the mode2 with one exception. The core is able to perform auto retuning during the transmission, so data length limitation is not exists. Mode 3 is currently not supported.  The driver can configure the timer by getting the  SDIO\_SRS17.RTNGCNT.  This field is to be 0 or 1, because the mode 3 is  not supported. |
| 13 | HwInit | HwInit | Use Tuning for SDR50  1 – tuning operation is necessary in SDR50 mode  0 – tuning operation is not necessary in SDR50  mode |
| 11:8 | HwInit | HwInit | Timer Count for Re-Tuning  These bits contain initial value for timer used to  starting periodically Re-Tuning Operation.   |  |  | | --- | --- | | 0h | Re-Tuning Timer disabled | | 1h | 1 second | | … | … | | N | seconds | | … | … | | Bh | 1024 seconds | | Eh-Ch | Reserved | | Fh | Obtain this info in other way | |
| 6 | HwInit | HwInit | 1.8V Line Driver Type D Supported  1 – Driver Type D supported  0 – Driver Type D not supported |
| 5 | HwInit | HwInit | 1.8V Line Driver Type C Supported  1 – Driver Type C supported  0 – Driver Type C not supported |
| 4 | HwInit | HwInit | 1.8V Line Driver Type A Supported  1 – Driver Type A supported  0 – Driver Type A not supported |
| 2 | HwInit | HwInit | DDR50 Supported  1 – DDR50 mode supported  0 – DDR50 mode not supported |
| 1 | HwInit | HwInit | SDR104 Supported  1 – SDR104 mode supported  0 – SDR104 mode not supported |
| 0 | HwInit | HwInit | SDR50 Supported  1 – SDR50 mode supported  0 – SDR50 mode not supported |

SDIO\_SRS18

Address: Operational Base + offset (0x48)

Table 525.SDIO\_SRS18 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 23:16 | HwInit | HwInit | Maximum Current for 1.8V   |  |  | | --- | --- | | MC18 | Maximum current | | 0 | Host System has to obtain the current value via another method. | | 1 | 4mA | | 2 | 8mA | | … | … | | 255 | 1020mA | |
| 15:8 | HwInit | HwInit | Maximum Current for 3.0V   |  |  | | --- | --- | | MC30 | Maximum current | | 0 | Host System has to obtain the current value via another method. | | 1 | 4mA | | 2 | 8mA | | … | … | | 255 | 1020mA | |
| 7:0 | HwInit | HwInit | Maximum Current for 3.3V   |  |  | | --- | --- | | MC33 | Maximum current | | 0 | Host System has to obtain the current value via another method. | | 1 | 4mA | | 2 | 8mA | | … | … | | 255 | 1020mA | |

SDIO\_SRS20

Address: Operational Base + offset (0x50)

Table 526.Event Trigger Register Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 25 | WO  (Write only) | 0 | Force ADMA Error Event. |
| 24 | WO | 0 | Force Auto CMD12 Error Event. |
| 23 | WO | 0 | Force Current Limit Error Event. |
| 22 | WO | 0 | Force Data End Bit Error Event. |
| 21 | WO | 0 | Force Data CRC Error Event. |
| 20 | WO | 0 | Force Data Timeout Error Event. |
| 19 | WO | 0 | Force Command Index Error Event. |
| 18 | WO | 0 | Force Command End Bit Error Event. |
| 17 | WO | 0 | Force Command CRC Error Event. |
| 16 | WO | 0 | Force Command Timeout Error Event. |
| 7 | WO | 0 | Force Command Not Issued By Auto CMD12 Error  Event. |
| 4 | WO | 0 | Force Auto CMD12 Index Error Event. |
| 3 | WO | 0 | Force Auto CMD12 End Bit Error Event. |
| 2 | WO | 0 | Force Auto CMD12 CRC Error Event. |
| 1 | WO | 0 | Force Auto CMD12 Timeout Error Event. |
| 0 | WO | 0 | Force Auto CMD12 Not Executed Event. |

SDIO\_SRS21

Address: Operational Base + offset (0x54)

Table 527.SDIO\_SRS21 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 2 | RW | 0 | ADMA Length Mismatch Error  This bit is set when:   total data length specified in ADMA descriptors  is different from that specified by the Block Count and Block Length fields (if Block Count Enable is set).   total data length cannot be divided by the block length (if Block Count Enable is not set). |
| 1:0 | RW | 0 | ADMA Error State  The value of this field reflects the state of the ADMA state machine. The possible values are:   |  |  | | --- | --- | | Value | State | | 2’b00 | ST\_STOP – ADMA Stopped | | 2’b01 | ST\_FDS – Fetching descriptor | | 2’b10 | not used | | 2’b11 | ST\_TRF – Transfer data | |

SDIO\_SRS22

Address: Operational Base + offset (0x58)

Table 528.ADMA System Address Register Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:0 | RW | 0 | ADMA System Address  The register holds the physical address of the  currently processed ADMA descriptor. The Host Driver will set this register with the descriptors table base address before it starts the ADMA transfers. The Host Driver should not write this register while the data transfer is active. While the ADMA engine is processing the descriptors list, the ADMASA value is always incremented to point the next descriptor to be fetched.  If the ADMA Error occurs, the register holds the  descriptor address depending on the ADMA Error  State (SDIO\_SRS21.ADMAES) register value, as listed in  the table below:   |  |  | | --- | --- | | ADMAES | State | | 2’b00 | Points next of the error descriptor | | 2’b01 | Points the error descriptor | | 2’b10 | not used | | 2’b11 | Points next of the error descriptor | |

SDIO\_SRS24(INT/DS)

Address: Operational Base + offset (0x60)

Table 529.SDIO\_SRS24 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 15:14/  31:0 | HwInit | HwInit | Driver Strength Select Value.This fields contains default value for SDIO\_SRS15.21..20 (Driver Strength Select). |
| 13:11/  29:27 | HwInit | HwInit | Clock Generator Select Value.This field contains default value for SDIO\_SRS11.5 (Clock Generator Select). |
| 9:0/  27:16 | HwInit | HwInit | SDCLK Frequency Select Value.This 9..8 and 7..0 field contain predefined value for SDIO\_SRS11.7..6 and SDIO\_SRS11.15..8, respectively.  ( {SDIO\_SRS11.7..6,SDIO\_SRS11.15..8} = SDCLKFSV) |

SDIO\_SRS25(HS/SDR12)

Address: Operational Base + offset (0x64)

Table 530.SDIO\_SRS25 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 15:14/  31:0 | HwInit | HwInit | Driver Strength Select Value  This fields contains default value for SDIO\_SRS15.21..20 (Driver Strength Select). |
| 13:11/  29:27 | HwInit | HwInit | Clock Generator Select Value.This field contains default value for SDIO\_SRS11.5 (Clock Generator Select). |
| 9:0/  27:16 | HwInit | HwInit | SDCLK Frequency Select Value .This 9..8 and 7..0 field contain predefined value for SDIO\_SRS11.7..6 and SDIO\_SRS11.15..8, respectively. ( {SDIO\_SRS11.7..6,SDIO\_SRS11.15..8} = SDCLKFSV) |

SDIO\_SRS26(SDR25/SDR50)

Address: Operational Base + offset (0x68)

Table 531.SDIO\_SRS26 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 15:14/  31:0 | HwInit | HwInit | Driver Strength Select Value  This fields contains default value for SDIO\_SRS15.21..20 (Driver Strength Select). |
| 13:11/  29:27 | HwInit | HwInit | Clock Generator Select Value. This field contains default value for SDIO\_SRS11.5 (Clock Generator Select). |
| 9:0/  27:16 | HwInit | HwInit | SDCLK Frequency Select Value.This 9..8 and 7..0 field contain predefined value for SDIO\_SRS11.7..6 and SDIO\_SRS11.15..8,respectively. ( {SDIO\_SRS11.7..6,SDIO\_SRS11.15..8} = SDCLKFSV) |

SDIO\_SRS27(SDR104/DDR50)

Address: Operational Base + offset (0x6C)

Table 532.SDIO\_SRS27 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 15:14/  31:0 | HwInit | HwInit | Driver Strength Select Value.This fields contains default value for SDIO\_SRS15.21..20 (Driver Strength Select). |
| 13:11/  29:27 | HwInit | HwInit | Clock Generator Select Value.This field contains default value for SDIO\_SRS11.5 (Clock Generator Select). |
| 9:0/  27:16 | HwInit | HwInit | SDCLK Frequency Select Value.This 9..8 and 7..0 field contain predefined value for SDIO\_SRS11.7..6 and SDIO\_SRS11.15..8, respectively.  ( {SDIO\_SRS11.7..6,SDIO\_SRS11.15..8} = SDCLKFSV) |

SDIO\_SRS56

Address: Operational Base + offset (0xE0)

Table 533.SDIO\_SRS56 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 30:24 | RW | 0 | Back-End Power Control  This register is to power control for the embedded  memory devices. This allows switching the memory between active and sleep modes.   |  |  | | --- | --- | | Bit No. | Backend power control for | | 24 | Device1 | | 25 | Device2 | | 26 | Device3 | | 27 | Device4 | | 28 | Device5 | | 29 | Device6 | | 30 | Device7 | |
| 22:20 | RW | 0 | Interrupt Pin Select  This register is to enable (1) / disable (0) interrupts INT\_A, INT\_B and INT\_C.Bit 22,21 and 20 are responsible for INT\_C, INT\_B and INT\_A, responsible. |
| 18:16 | RW | 0 | Clock Pin Select  The active device on the shared bus can be selected by this register. The selected device is only supplied with sdclk, so only this device will receive the commands from host.   |  |  | | --- | --- | | Value | Selected clock pins | | 0 | None | | 1 | sdclk for device 1 | | 2 | sdclk for device 2 | | 3 | sdclk for device 3 | | 4 | sdclk for device 4 | | 5 | sdclk for device 5 | | 6 | sdclk for device 6 | | 7 | sdclk for device 7 | |
| 14:8 | HwInit | HwInit | Bus Width Preset  This field contains predefined bus width for the  shared bus devices. Each of devices will use either  4-bit or 8-bit DAT line. The 1-bit DAT bus is not  applicable for devices on shared bus.  The 4-bit bus is encoded as 0, while 8-bit bus is  encoded as 1   |  |  | | --- | --- | | Bit No. | Bus Width for | | 8 | Device 1 | | 9 | Device 2 | | 10 | Device 3 | | 11 | Device 4 | | 12 | Device 5 | | 13 | Device 6 | | 14 | Device 7 |   The field is valid only when Slot Type is set 2.In other case filed is to be ignored. |
| 5:4 | HwInit | HwInit | Number of Interrupt Input Pins  This field inform about supported interrupt input pins. Any device connected to the shared bus can use any of three interrupt lines.  Multiple devices can be connected to the single interrupt through wired-AND connection.   |  |  | | --- | --- | | 0 | Interrupt pins are not supported | | 1 | INT\_A is supported | | 2 | INT\_A and INT\_B are supported | | 3 | INT\_A, INT\_B and INT\_C are supported | |
| 2:0 | HwInit | HwInit | Number of Clock Pins  This field gives information about shared bus  support and a number of the shared devices  connected to the single SD interface. The host  supports up to 7 devices.   |  |  | | --- | --- | | 0 | Shared bus not supported | | 1 | Support for 1 sdclk clock | | 2 | Support for 2 sdclk clocks | | … | … | | 7 | Support for 7 sdclk clocks. | |

CRS63

Address: Operational Base + offset (0xFC)

Table 534.CRS63 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | HwInit | HwInit | Vendor Version Number  This field is reserved for the vendor version number. The Host Driver would not use this status. |
| 23:16 | HwInit | HwInit | Specification Version Number  This field identifies the Host Controller Specification Version.  SVN should be 02h for current version of the SDIO- HOST, indicating compatibility with SD Host Specification Version 3.00. |
| 3:0 | ROC | 0 | Interrupt Signal For Each Slot.  These status bits indicate the logical OR of  Interrupt Signal and Wakeup Signal for each slot.  CRS63.0 – slot #0 interrupt status  CRS63.1 – slot #1 interrupt status  CRS63.2 – slot #2 interrupt status |

* 1. Programming Sequence
     1. Card Initialization

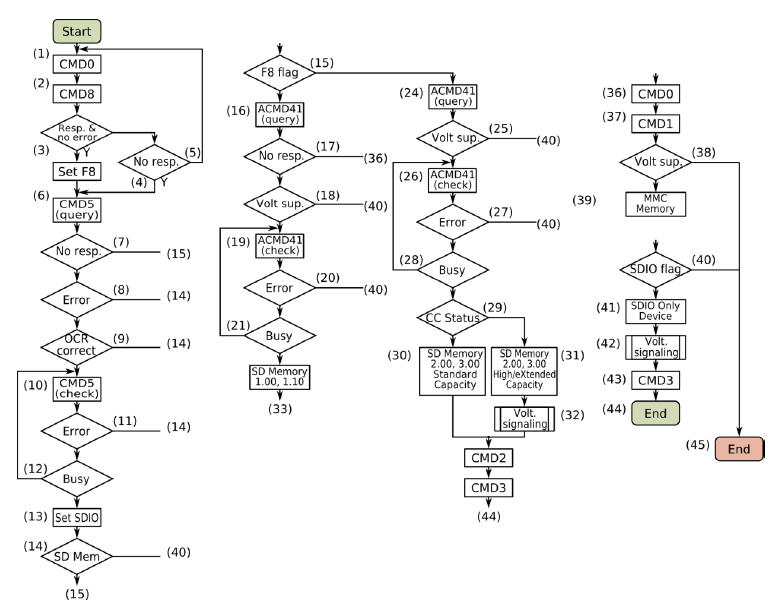


Table 535.Card Initialization chart

Card initialization procedure can be performed by going throughout following steps:

(1) First, the card is reset by CMD0.

(2) Send CMD8.

(3) If the response is correct, set software F8 flag.

(4) If response is not received, clear software F8 flag.

(5) If any error occurs during the command-response transfer or any error is detected within response content, reset the procedure. Go to step (1). If the problem is repeating for the card, the software can disconnect the card.

**== SDIO Initialization ==**

(6) Software reads OCR from SDIO by querying card with CMD5.

(7) If response is not received card not offers SDIO functionality. Go to step (15).

(8) If any response error occurs, skip SDIO functionality initialization Go to step (14).

(9) If SDIO functions’ OCR voltage range has no common range with host then do not initialize SDIO functionality. Go to step (14).

(10) Send the CMD5 with host voltage range and request of low voltage signaling request (UHS-I option).

(11) If the response is not received or response is received with error, skip SDIO initialization and go to (14).

(12) If card response with busy status, go to (10). If busy period is longer than 1 second, skip SDIO initialization, go to (13).

(13) Set software SDIO flag.

(14) The responses get in steps (6) and (9) gives information whether card supports the SD Memory. If does not support then skip SD Memory initialization, go to (40).

**== SD Memory (1.xx) Initialization ==**

(15) Check software flag F8. If flag is set go to step (24).

(16) Send ACMD41 query to obtain the OCR.

(17) If no response is received then go to step (36).

(18) If memory device does not support the host’s voltage range, go to (40).

(19) Send ACMD41 with voltage range supported by host.

(20) If response error occurs, go to (40).

(21) If SD memory device is busy, go to step (19). If busy timeout occurs then go to step (40).

(22) The detected card is SD Standard Capacity card ver. 1.00 or 1.10.

(23) Go to the step (33).

**== SD Memory (2.00 and above) Initialization ==**

(24) Send ACMD41 query to obtain the OCR.

(25) If memory device does not support the host’s voltage range, go to (40).

(26) Send ACMD41 with voltage range supported by host, High Capacity selected, and 1.8V signaling request.

(27) If response error occurs, go to (40).

(28) If SD memory device is busy, go to (26). If busy timeout occurs then go to step (40).

(29) If Card Capacity Status is set go to step (31).

(30) The detected card is SD Standard Capacity card ver. 2.00 or 3.00. Go to step (33).

(31) The detected card is SD High/eXtended Capacity card.

(32) Perform Signaling Voltage Switch procedure.

**== Memory device continuation ==**

(33) Send CMD2 to obtain CID.

(34) Send CMD3 to obtain RCA.

(35) Go to step (44)

**== MMC Memory ==**

(36) Send CMD0 to reset the card.

(37) Send CMD1.

(38) If the card’s voltage range is incompatible with the hosts voltage range then disable the card and go to step (45).

(39) The MMC memory is detected, go to step (33).

**== SDIO device continuation ==**

(40) If the SDIO flag is cleared, then card is not recognized, and should be disconnected. Go to step (45).

(41) The SDIO-only card is detected.

(42) Perform Signal Voltage Switch procedure.

(43) Send CMD3 to obtain RCA.

**== Device Initialization Success ==**

(44) The card has been successfully detected. Stop

**== Device Initialization Failure ==**

(45) The card is not recognized or not compatible. It should be disconnected. The software/host should not perform any future access to the card. Stop.

* + 1. Voltage Switching Sequence (UHS-I only)

The following sequence is performed to change the interface signaling level between 3.3V and 1.8V for UHS mode supporting devices. The signaling level is changed on the on following SD lines: clk, cmd and dat.

The 3.3V is used when Default Speed or High Speed mode are used. The 1.8V is used when the any of the UHS-I mode will be used.



Figure 34.Voltage Switching chart

Follows the steps to perform the Voltage Switching procedure:

(1) Check card’s response field for S18A flag(Switch to 1.8V Accepted) This flag can be found in response to the ACMD41 (SD Memory) and CMD5 (SDIO).

(2) The S18A=0 means that the host shall remain in the 3.3V signaling level. Voltage switching is not performed. Stop.

(3) Send CMD11 command to initialize voltage switching

(4) If no response is received (response timeout occurs), go to step (17).

(5) Disable SDCLK by writing 0 to SRS11.2/SDCE.

(6) Check if the DAT[3:0] lines are asserted to 0000b by the card.

(7) If the previous step fails, go to step (17).

(8) Set host signaling voltage level to 1.8V by writing 1 to the SRS15.19/1V8SE.

(9) Wait for 5ms.

(10) Check if the SRS15.19/1V8SE is enabled.

(11) If previous step failed, go to step (17).

(12) Enable SDCLK by writing 1 to SRS11.2/SDCE.

(13) Wait for 1ms.

(14) Check if the DAT[3:0] lines are 1111b.

(15) If previous step failed, go to step (17).

(16) Procedure succeeds; stop.

(17) Procedure fails; disable the power supply (by writing 0 to SRS10.8/BP) and stop.

* + 1. Clock Tuning procedure (UHS-I only)

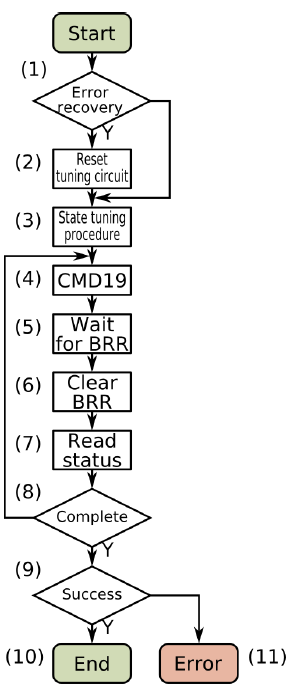


Figure 35.Clock Tuning chart

Tuning procedure is used when the sampling clock is to be adjusted to the data (CMD/DAT lines) in SDR104 (SDR50 optionally). Follows the instructions below to perform the procedure:

(1) If error recovery is not required, go to step (3).

(2) Reset tuning circuit by writing 0 to SRS15.23/SCS.

(3) Start tuning procedure by writing 1 to SRS15.22/EXTNG.

(4) Send CMD19.

(5) Wait for Buffer Read Ready (SRS12.5/BRR) equal 0.

(6) Clear SRS12.5/BRR.

(7) Read the SRS15.22/EXTNG register.

(8) If the procedure is not complete (SRS15.22/EXTNG = 0) go to step (4).

(9) Check procedure result by reading the SRS15.23/SCS. If the SRS15.23/SCS=0 go to (11)

(10) Sampling clock is tuned up. Procedure ends with success. Stop.

(11) Tuning failed. Stop.

* + 1. Commands Which Do Not Use DAT Line

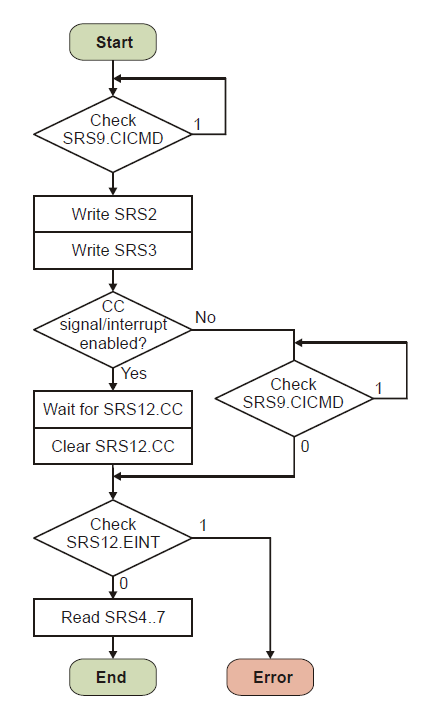


Figure 36.Commands Which Do Not Use DAT Line

Commands in this category include all commands other than commands which transfer data through DAT line or commands with a BUSY response.

Before sending any command, the software has to verify that no other command/response cycle is currently in progress by checking SRS9.CICMD bit (Command Inhibit CMD). When SRS9.CICMD = 0, then the CMD line is idle and the software is free to send another command. First, the software writes an appropriate command argument, 32-bit field in SRS2. Then it writes SRS3 with an appropriate command configuration. Writing SRS3.CI (Command Index) field directly triggers the command generation/response checking process.

When command generation/response checking is in progress, the SRS9.CICMD is set to 1. After finishing the entire command/response cycle on the SD interface, SRS9.CICMD is automatically cleared by the SDIO-HOST. High to low transition on SRS9.CICMD triggers the Command Complete interrupt (SRS12.CC) if it is enabled (SRS13.CC\_SE & SRS14.CC\_IE enable bits).

At this time, the command/response cycle is complete and the software can read the card response from SRS.4 – SRS.7. The software can also check for possible response error status bits (SRS12.ECI, SRS12.ECEB, SRS12.ECCRC, SRS12.ECT).

After finishing with the command, the software clears the Command Complete Interrupt (SRS12.CC) by writing 1 to this bit.

* + 1. Commands with BUSY

Commands in this category use the DAT line for BUSY signaling. The procedure is very similar to the procedure of sending the commands which do not use DAT line. The difference is that, after checking SRS9.CICMD bit (Command Inhibit CMD), the software also checks SRS9.CIDAT (Command Inhibit DAT). The command with BUSY can only be sent if both SRS9.CICMD and SRS9.CIDAT are cleared. The only exception is the ABORT type command, that is, a command with SRS3.CT (Command Type) = 11b. The ABORT command can be sent even if SRS9.CIDAT = 0. It is possible, since the ABORT commands are used for breaking the other transaction that uses DAT line.

When command generation/response checking is in progress, the SRS9.CICMD is set to 1. When sending the command on CMD, SDIO-HOST also sets SRS9.CIDAT to 1. After finishing the command/ response cycle on the CMD line, SRS9.CICMD is automatically cleared by the SDIO-HOST. High to low transition on SRS9.CICMD triggers the Command Complete interrupt (SRS12.CC) if it is enabled (SRS13.CC\_SE & SRS14.CC\_IE enable bits). The software will clear SRS12.CC manually by writing logical 1.

After the command/response cycle on CMD, the card can send the BUSY status using the DAT line. SRS9.CIDAT remains set until the BUSY is released. High to low transition on SRS9.CIDAT triggers the Transfer Complete interrupt (SRS12.TC) if it is enabled (SRS13.TC\_SE & SRS14.TC\_IE enable bits). The software will wait until the BUSY is released by looking at the SRS9.CIDAT or, alternatively, by waiting for SRS12.TC interrupt. After the Transfer Complete event, the SRS12.TC will be cleared by the software. At this time, the entire command/response cycle is complete.

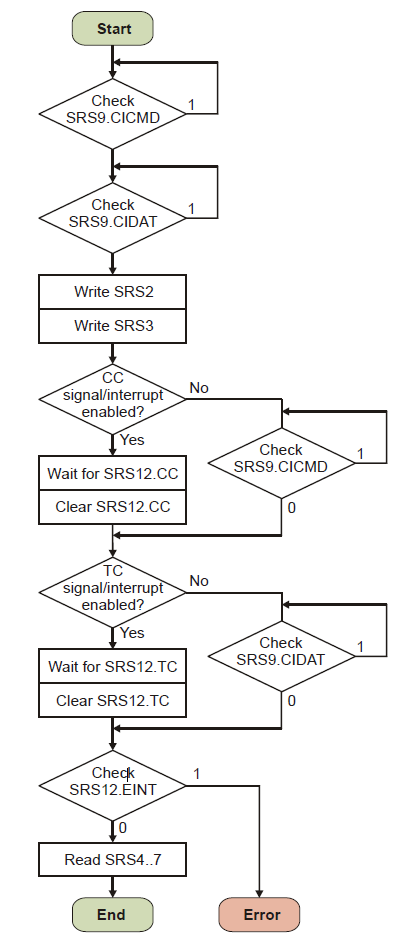


Figure 37.Commands with BUSY

* + 1. Commands Which Transfer Data on the DAT Line without DMA

Commands in this category use the DAT line for a data transfer. Such commands include single-block and multi-block read/write data transfers. The read transfer term indicates that the data is read from the card to the SDIO-HOST. The write transfer indicates that the data is written to the card by the SDIO-HOST.

The software first checks the Command Inhibit CMD (SRS9.CICMD) and Command Inhibit DAT (SRS9.CIDAT) before doing anything else, exactly as in the case of the commands with BUSY. The data transfer command can only be started when SRS9.CICMD = SRS9.CIDAT = 0.

The data transfer requires the appropriate configuration setup in SRS1 register: SRS1.BCCT (Block Count for Current Transfer), and SRS1.TBS (Transfer Block Size). Also SRS3.BCE is set for the transfer with a known number of data blocks, or cleared for the infinite data transfer.

A few further steps are common for all command types. The software writes an appropriate command argument in SRS2. Then it writes SRS3 with the command configuration. SRS3.DPS (Data Present Select) is set to 1. SRS3.DTDS (Data Transfer Direction Select) is 1 for the read transfers, or 0 for the write transfers. SRS3.DMAE is cleared, since the DMA is not used.

When command generation/response checking is in progress, the SRS9.CICMD is set to 1. After finishing the entire command/response cycle on the SD interface, SRS9.CICMD is automatically cleared by the SDIO-HOST. High to low transition on SRS9.CICMD triggers the Command Complete interrupt (SRS12.CC) if it is enabled. The software will clear SRS12.CC manually by writing logical 1.

Then the software will process the data blocks.

In case of the write transfer, the data would be written to SRS8 (Data Buffer) on a block by block basis. SRS9.BWE (Buffer Write Enable) can be used to check if there is a free FIFO buffer inside SDIO-HOST. SRS9.BWE = 1 means that there is enough space to write the entire block. Alternatively, the software can use SRS12.BWR (Buffer Write Ready) interrupt that is triggered whenever the FIFO buffer becomes ready for the transfer.

In case of the read transfer, the data would be read from SRS8 (Data Buffer) on a block by block basis. SRS9.BRE (Buffer Read Enable) indicates that the entire block can be read from the FIFO. Alternatively, the SRS12.BRR (Buffer Read Ready) is generated whenever the SRS9.BRE changes from 0 to 1, indicating the new data block ready in the FIFO.

The software will process all data blocks as indicated by the SRS1.BCCT (Block Count for Current Transfer). During the entire data transfer, SRS9.CIDAT (Command Inhibit DAT) remains set. SRS9.CIDAT is cleared by the SDIO-HOST after the last block transfers to/from the FIFO. High to low transition on SRS9.CIDAT triggers the Transfer Complete interrupt (SRS12.TC) if it is enabled (SRS13.TC\_SE & SRS14.TC\_IE enable bits).

The software will wait until the busy is released by looking at the SRS9.CIDAT or, alternatively, by waiting for SRS12.TC interrupt. After the Transfer Complete event, the SRS12.TC will cleared by the software. The only exception is for the infinite data transfers. The transfer is infinite when SRS3.BCE (Block Count Enable) = 0. In this case, the block count is not known to the SDIO-HOST, and the software has to break the transfer manually by performing the Abort procedure. This Abort procedure is described in a separate section.

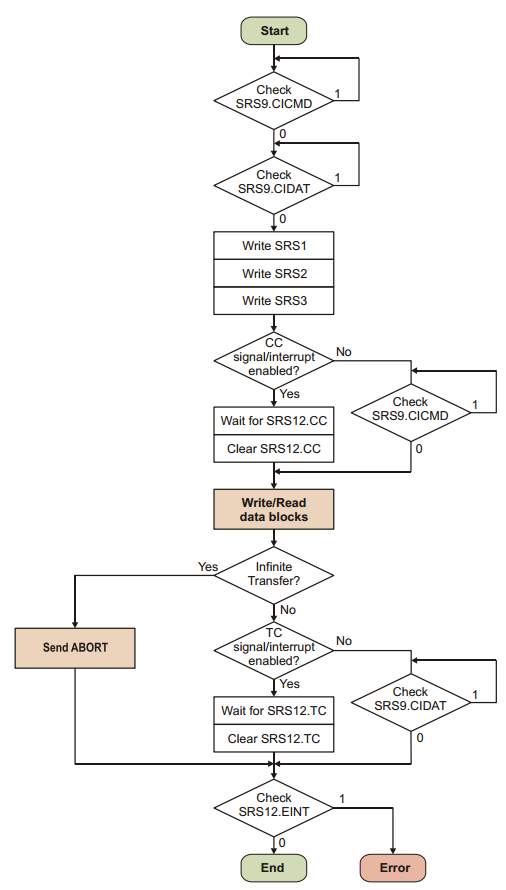


Figure 38.Commands Which Transfer Data without DMA

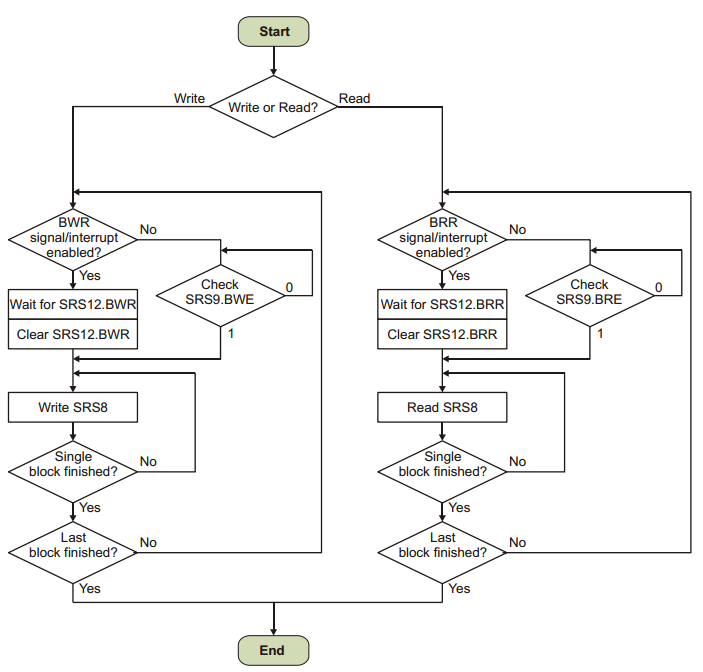


Figure 39.Write/Read Data Blocks Sub-procedure

* + 1. Commands Which Transfer Data on the DAT Line Using DMA

This entire procedure is based on the previous procedure (for commands which do not use DMA). There are some differences in the setup:

SRS0.SA (System Address) is written with the memory address of the first data block. All data blocks are ready in the system memory (will contain valid data in case of write transfer, or will be reserved for the read transfer).

SRS1.HDMABB (Host DMA Address Buffer Boundary) is written with the appropriate value (this value depends on the system memory and bus architecture).

SRS3.DMAE (DMA Enable) bit is set to 1.

SRS3.BCE (Block Count Enable) is set to 1. This is because the infinite transfers are not possible when using DMA.

SRS13.DINT\_SE (DMA Interrupt Signal Enable) is set to 1.

When DMA is enabled, the software does not have to process the data buffer manually. The SDIO-HOST will transfer the data between the system and the internal FIFO automatically, block by block. After transferring the last data block, the SDIO-HOST clears SRS9.CIDAT (Command Inhibit DAT), and generates SRS12.TC (Transfer Complete Interrupt) if it is enabled.

During the data transaction, the SDIO-HOST also generates SRS12.DINT (DMA Interrupt) when crossing the boundary defined in SRS1.HDMABB (Host DMA Buffer Boundary). In such a case, the software would clear SRS1.DINT by writing 1, and restart the DMA operation by writing the SRS0.SA register. The SDIO-HOST restarts the DMA transfer from the current SRS0.SA value.

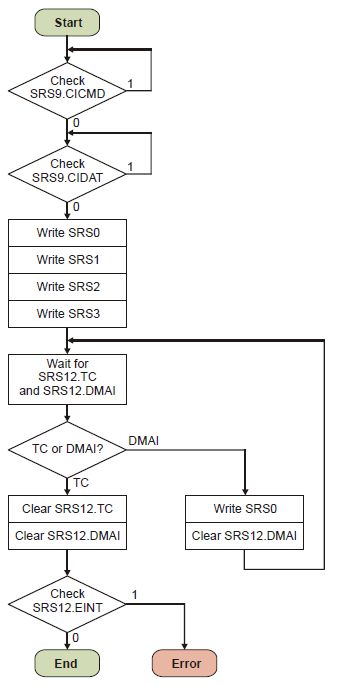


Figure 40.Commands Which Transfer Data Using DMA

1. DDR Controller + PHY
   1. Register Summary

Table 536.DDR Controller&PHY Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Offset** | **Reset Value** | **Description** |
| **Controller Register** | | | |
| DENALI\_CTL\_00 | 0x000 | 0x20410000 | Version, DRAM class and start command setting. |
| DENALI\_CTL\_01 | 0x004 | 0x00020b10 | FiFo depth, max width of Colum address and memory max address width information. |
| DENALI\_CTL\_02 | 0x008 | 0x00000000 | Write data fifo pointer width, write data fifo path, async cdc stages information. |
| DENALI\_CTL\_03 | 0x00C | 0x00000000 | AXI0 fifo and array information |
| DENALI\_CTL\_04 | 0x010 | 0x00000000 | AXI1 fifo and array information |
| DENALI\_CTL\_05 | 0x014 | 0x00000000 | AXI1 fifo and array information |
| DENALI\_CTL\_06 | 0x018 | 0x00000000 | obsolete |
| DENALI\_CTL\_07 | 0x01C | 0x00000000 | obsolete |
| DENALI\_CTL\_08 | 0x020 | 0x00000000 | DRAM TINT register |
| DENALI\_CTL\_09 | 0x024 | 0x00000000 | DRAM TINT3 register |
| DENALI\_CTL\_10 | 0x028 | 0x00000000 | DRAM TINT4 register |
| DENALI\_CTL\_11 | 0x02C | 0x00000000 | Auto MRR init and DRAM TINT5 setting |
| DENALI\_CTL\_12 | 0x030 | 0x00000000 | obsolete |
| DENALI\_CTL\_13 | 0x034 | 0x00000000 | obsolete |
| DENALI\_CTL\_14 | 0x038 | 0x00000000 | DRAM burst interrupt interval value |
| DENALI\_CTL\_15 | 0x03C | 0x00000000 | DRAM TRAS\_MIN, TRC, TRRD, CAS-to-CAS value setting |
| DENALI\_CTL\_16 | 0x040 | 0x00000000 | DRAM TRTP, TFAW, TRP, TWTR value setting |
| DENALI\_CTL\_17 | 0x044 | 0x00000000 | TMOD and TMRD setting |
| DENALI\_CTL\_18 | 0x048 | 0x00000000 | Min CKE pulse and DRAM TRAS\_MAX value setting |
| DENALI\_CTL\_19 | 0x04C | 0x00000000 | TWR, TRCD, WRITEINTERRUPT, and Min CKE pulse setting. |
| DENALI\_CTL\_20 | 0x050 | 0x00000000 | DRAM setting. |
| DENALI\_CTL\_21 | 0x054 | 0x00000000 | Auto pre-change mode and DRAM setting. |
| DENALI\_CTL\_22 | 0x058 | 0x02000000 | Burst length enable, DRAM TDAL, transition lock out, and allows the controller to issue command. |
| DENALI\_CTL\_23 | 0x05C | 0x00000000 | Auto-refresh, DIMM, DRAM TRP all blank setting. |
| DENALI\_CTL\_24 | 0x060 | 0x00000000 | N/A |
| DENALI\_CTL\_25 | 0x064 | 0x00000000 | DRAM TREF value. |
| DENALI\_CTL\_26 | 0x068 | 0x00000000 | DRAM TPDEX value. |
| DENALI\_CTL\_27 | 0x06C | 0x00000000 | Port command error type and ID |
| DENALI\_CTL\_28 | 0x070 | 0x00000000 | DRAM TXSNR and TXSR setting |
| DENALI\_CTL\_29 | 0x074 | 0x00000000 | CKE, ENABLE\_QUICK\_SREFRESH, PWR\_SEREFRESH\_EXIT value setting. |
| DENALI\_CTL\_30 | 0x078 | 0x00000000 | LP\_CMD, CKSRX, CKSRE, and LOPOWER\_REFRESH\_ENABLE setting. |
| DENALI\_CTL\_31 | 0x07C | 0x00000000 | LPI\_TIMER\_WAKEUP, LPI\_SR\_MCCLK\_GATE\_WAKEUP, LPI\_SR\_WAKEUP and LPI\_PD\_WAKEUP setting. |
| DENALI\_CTL\_32 | 0x080 | 0x00000000 | LPI\_TIMER\_COUNT and LPI\_WAKEUP\_EN setting. |
| DENALI\_CTL\_33 | 0x084 | 0x20000000 | LP\_STATE, TDFI\_LP\_RESP and LPI\_WAKEUP\_TIMEOUT setting. |
| DENALI\_CTL\_34 | 0x088 | 0x00000000 | LP\_AUTO\_MEM\_GATE\_EN, LP\_AUTO\_EXIT\_EN, LP\_AUTO\_ENTRY\_EN setting and LP\_ARB\_STATE information. |
| DENALI\_CTL\_35 | 0x08C | 0x00000000 | LP\_AUTO\_SR\_MC\_GATE\_IDLE, LP\_AUTO\_SR\_IDLE and LP\_AUTO\_PD\_IDLE setting. |
| DENALI\_CTL\_36 | 0x090 | 0x00000000 | Reserved |
| DENALI\_CTL\_37 | 0x094 | 0x00000000 | Write memory mode register data to the DRAMs |
| DENALI\_CTL\_38 | 0x098 | 0x00000000 | READ\_MODEREG setting and MRW\_STATUS information. |
| DENALI\_CTL\_39 | 0x09C | 0x00000000 | Data and chip returned from memory mode register read requested by the READ\_MODEREG parameter. |
| DENALI\_CTL\_40 | 0x0A0 | 0x00000000 | AUTO\_TEMPCHK\_VAL\_0 and PERIPHERAL\_MRR\_DATA [39:32] information. |
| DENALI\_CTL\_41 | 0x0A4 | 0x00000000 | Number of long count sequences between automatic memory mode register read commands of MR4 |
| DENALI\_CTL\_42 | 0x0A8 | 0x00000000 | Data to program into memory mode register 1 for chip select 0. |
| DENALI\_CTL\_43 | 0x0AC | 0x00000000 | MRSINGLE\_DATA\_0 and MR2\_DATA\_0 setting. |
| DENALI\_CTL\_44 | 0x0B0 | 0x00000000 | MR11\_DATA\_0 and MR3\_DATA\_0 setting. MR8\_DATA\_0 information. |
| DENALI\_CTL\_45 | 0x0B4 | 0x00000000 | MR17\_DATA\_0 and MR16\_DATA\_0 setting. |
| DENALI\_CTL\_46 | 0x0B8 | 0x00000000 | Obsolete |
| DENALI\_CTL\_47 | 0x0BC | 0x00000000 | Obsolete |
| DENALI\_CTL\_48 | 0x0C0 | 0x00000000 | Obsolete |
| DENALI\_CTL\_49 | 0x0C4 | 0x00000000 | Start BIST checking at this address |
| DENALI\_CTL\_50 | 0x0C8 | 0x00000000 | BIST\_ADDR\_CHECK and BIST\_DATA\_CHECK setting. |
| DENALI\_CTL\_51 | 0x0CC | 0x00000000 | Start BIST checking at this address |
| DENALI\_CTL\_52 | 0x0D0 | 0x00000000 | Obsolete |
| DENALI\_CTL\_53 | 0x0D4 | 0x00000000 | Mask applied to data for BIST error checking. |
| DENALI\_CTL\_54 | 0x0D8 | 0x00000000 | Mask applied to data for BIST error checking. |
| DENALI\_CTL\_55 | 0x0DC | 0x00000000 | ZQINIT and LONG\_COUNT\_MASK setting. |
| DENALI\_CTL\_56 | 0x0E0 | 0x00000000 | ZQCS and ZQCL setting. |
| DENALI\_CTL\_57 | 0x0E4 | 0x00000000 | ZQ\_ON\_SREF\_EXIT and ZQ\_REQ setting. |
| DENALI\_CTL\_58 | 0x0E8 | 0x00000000 | Number of long count sequences allowed between automatic ZQCS commands |
| DENALI\_CTL\_59 | 0x0EC | 0x00000000 | NO\_ZQ\_INIT, ZQRESET setting and ZQ\_IN\_PROGRESS information. |
| DENALI\_CTL\_60 | 0x0F0 | 0x00000000 | COL\_DIFF, ROW\_DIFF and BANK\_DIFF setting |
| DENALI\_CTL\_61 | 0x0F4 | 0x00000000 | AGE\_COUNT, APREBIT, BANK\_ADDR\_INTLV\_EN and BANK\_START\_BIT setting. |
| DENALI\_CTL\_62 | 0x0F8 | 0x00000000 | BANK\_SPLIT\_EN, ADDR\_CMP\_EN and COMMAND\_AGE\_COUNT setting. |
| DENALI\_CTL\_63 | 0x0FC | 0x00000000 | RW\_SAME\_PAGE\_EN, RW\_SAME\_EN, PRIORITY\_EN and PLACEMENT\_EN setting. |
| DENALI\_CTL\_64 | 0x100 | 0x00000000 | NUM\_Q\_ENTRIES\_ACT\_DISABLE and DISABLE\_RW\_GROUP\_W\_BNK\_CONFLICT setting. |
| DENALI\_CTL\_65 | 0x104 | 0x00000000 | INHIBIT\_DRAM\_CMD, DISABLE\_RD\_INTERLEAVE and SWAP\_EN setting. |
| DENALI\_CTL\_66 | 0x108 | 0x00000000 | MEMDATA\_RATIO\_0, LPDDR2\_S4, and REDUC setting. |
| DENALI\_CTL\_67 | 0x10C | 0x00000000 | WR\_ORDER\_REQ, IN\_ORDER\_ACCEPT, and Q\_FULLNESS setting. |
| DENALI\_CTL\_68 | 0x110 | 0x00000000 | CTRLUPD\_REQ\_PER\_AREF\_EN, CTRLUPD\_REQ setting and DFI\_ERROR, CONTROLLER\_BUSY information. |
| DENALI\_CTL\_69 | 0x114 | 0x00000000 | DFI error information |
| DENALI\_CTL\_70 | 0x118 | 0x00000000 | Interrupt status |
| DENALI\_CTL\_71 | 0x11C | 0x00000000 | Interrupt Ack |
| DENALI\_CTL\_72 | 0x120 | 0x00000000 | Interrupt mask |
| DENALI\_CTL\_73 | 0x124 | 0x00000000 | Out of range address |
| DENALI\_CTL\_74 | 0x128 | 0x00000000 | Out of range source ID |
| DENALI\_CTL\_75 | 0x12C | 0x00000000 | Bist except data error |
| DENALI\_CTL\_76 | 0x130 | 0x00000000 | Bist except data error |
| DENALI\_CTL\_77 | 0x134 | 0x00000000 | Bist except data error |
| DENALI\_CTL\_78 | 0x138 | 0x00000000 | Bist except data error |
| DENALI\_CTL\_79 | 0x13C | 0x00000000 | Bist except data error |
| DENALI\_CTL\_80 | 0x140 | 0x00000000 | Data on Bist error |
| DENALI\_CTL\_81 | 0x144 | 0x00000000 | Data on Bist error |
| DENALI\_CTL\_82 | 0x148 | 0x00000000 | Data on Bist error |
| DENALI\_CTL\_83 | 0x14C | 0x00000000 | Data on Bist error |
| DENALI\_CTL\_84 | 0x150 | 0x00000000 | Address of Bist error |
| DENALI\_CTL\_85 | 0x154 | 0x00000000 | Obsolete |
| DENALI\_CTL\_86 | 0x158 | 0x00000000 | Address of command that caused the PORT command error. |
| DENALI\_CTL\_87 | 0x15C | 0x00000000 | Port command error type and ID |
| DENALI\_CTL\_88 | 0x160 | 0x00000000 | Obsolete |
| DENALI\_CTL\_89 | 0x164 | 0x00000000 | ODT\_EN, TODTH\_WR and TODTL\_2CMD setting. |
| DENALI\_CTL\_90 | 0x168 | 0x01000000 | Enable controller to assert ODT at all times except during reads. |
| DENALI\_CTL\_91 | 0x16C | 0x00010101 | Additional delay to insert between two reads to the same chip select. Any value including 0x0 supported. |
| DENALI\_CTL\_92 | 0x170 | 0x00000002 | TDQSCK\_MAX, W2W\_SAMECS\_DLY, W2R\_SAMECS\_DLY and R2W\_SAMECS\_DLY setting. |
| DENALI\_CTL\_93 | 0x174 | 0x00000000 | SWLVL\_START, SWLVL\_LOAD and SW\_LEVELING\_MODE setting. |
| DENALI\_CTL\_94 | 0x178 | 0x00000000 | SWLVL\_RESP\_1, SWLVL\_RESP\_0 and SWLVL\_OP\_DONE information. SWLVL\_EXIT setting. |
| DENALI\_CTL\_95 | 0x17C | 0x00000000 | WRLVL\_CS and WRLVL\_REQ setting.  SWLVL\_RESP\_3 and SWLVL\_RESP\_2 information. |
| DENALI\_CTL\_96 | 0x180 | 0x00000000 | WRLVL\_EN, WLMRD and WLDQSEN setting. |
| DENALI\_CTL\_97 | 0x184 | 0x00000000 | WRLVL\_PERIODIC, DFI\_PHY\_WRLVL\_MODE and WRLVL\_INTERVAL setting. |
| DENALI\_CTL\_98 | 0x188 | 0x00000000 | WRLVL\_AREF\_EN, WRLVL\_RESP\_MASK  WRLVL\_ON\_SREF\_EXIT setting. |
| DENALI\_CTL\_99 | 0x18C | 0x00000000 | RDLVL\_GATE\_REQ, RDLVL\_REQ setting and WRLVL\_ERROR\_STATUS information. |
| DENALI\_CTL\_100 | 0x190 | 0x00000000 | DFI\_PHY\_RDLVL\_MODE, RDLVL\_GATE\_SEQ\_EN, RDLVL\_SEQ\_EN and RDLVL\_CS setting. |
| DENALI\_CTL\_101 | 0x194 | 0x00000000 | RDLVL\_GATE\_PERIODIC, RDLVL\_ON\_SREF\_EXIT, RDLVL\_PERIODIC and DFI\_PHY\_RDLVL\_GATE\_MODE setting. |
| DENALI\_CTL\_102 | 0x198 | 0x00000000 | RDLVL\_AREF\_EN and RDLVL\_GATE\_ON\_SREF\_EXIT setting. |
| DENALI\_CTL\_103 | 0x19C | 0x00000000 | User request to initiate CA training. |
| DENALI\_CTL\_104 | 0x1A0 | 0x00000000 | CALVL\_PAT\_0 and CALVL\_CS setting. |
| DENALI\_CTL\_105 | 0x1A4 | 0x00000000 | CA Training pattern 0 driven on the CA bus before and after a calibration command. |
| DENALI\_CTL\_106 | 0x1A8 | 0x00000000 | CA Training pattern 1 driven on the CA bus during a calibration command. |
| DENALI\_CTL\_107 | 0x1AC | 0x00000000 | CA Training pattern 1 driven on the CA bus before and after a calibration command. |
| DENALI\_CTL\_108 | 0x1B0 | 0x00000000 | CA Training pattern 2 driven on the CA bus during a calibration command. |
| DENALI\_CTL\_109 | 0x1B4 | 0x00000000 | CA Training pattern 2 driven on the CA bus before and after a calibration command. |
| DENALI\_CTL\_110 | 0x1B8 | 0x00000000 | CA Training pattern 3 driven on the CA bus during a calibration command. |
| DENALI\_CTL\_111 | 0x1BC | 0x00000000 | CA Training pattern 3 driven on the CA bus before and after a calibration command. |
| DENALI\_CTL\_112 | 0x1C0 | 0x00000000 | CALVL\_PERIODIC, DFI\_PHY\_CALVL\_MODE and CALVL\_SEQ\_EN setting. |
| DENALI\_CTL\_113 | 0x1C4 | 0x00000000 | CALVL\_AREF\_EN and CALVL\_ON\_SREF\_EXIT setting. |
| DENALI\_CTL\_114 | 0x1C8 | 0x00000000 | AXI0\_FIFO\_TYPE\_REG, AXI0\_W\_PRIORITY, AXI0\_R\_PRIORITY and  AXI0\_FIXED\_PORT\_PRIORITY\_ENABLE setting. |
| DENALI\_CTL\_115 | 0x1CC | 0x00000000 | AXI1\_FIFO\_TYPE\_REG, AXI1\_W\_PRIORITY, AXI1\_R\_PRIORITY and  AXI1\_FIXED\_PORT\_PRIORITY\_ENABLE setting. |
| DENALI\_CTL\_116 | 0x1D0 | 0x00000000 | Obsolete |
| DENALI\_CTL\_117 | 0x1D4 | 0x00000000 | Obsolete |
| DENALI\_CTL\_118 | 0x1D8 | 0x00000000 | AXI0\_CURRENT\_BDW information.  AXI0\_BDW\_OVFLOW, AXI0\_BDW and ARB\_CMD\_Q\_THRESHOLD setting. |
| DENALI\_CTL\_119 | 0x1DC | 0x00000000 | AXI1\_CURRENT\_BDW information.  AXI1\_BDW\_OVFLOW and AXI1\_BDW setting. |
| DENALI\_CTL\_120 | 0x1E0 | 0x00000000 | Obsolete |
| DENALI\_CTL\_121 | 0x1E4 | 0x00000000 | CKE status |
| DENALI\_CTL\_122 | 0x1E8 | 0xXX000000 | TDFI\_PHY\_WRLAT information.  DLL\_RST\_ADJ\_DLY and DLL\_RST\_DELAY setting. |
| DENALI\_CTL\_123 | 0x1EC | 0x00XX0600 | DRAM\_CLK\_DISABLE and TDFI\_PHY\_RDLAT setting.  TDFI\_RDDATA\_EN and UPDATE\_ERROR\_STATUS information. |
| DENALI\_CTL\_124 | 0x1F0 | 0x00000000 | TDFI\_CTRLUPD\_MAX and TDFI\_CTRLUPD\_MIN setting. |
| DENALI\_CTL\_125 | 0x1F4 | 0x00000000 | TDFI\_PHYUPD\_TYPE1 and TDFI\_PHYUPD\_TYPE0 setting. |
| DENALI\_CTL\_126 | 0x1F8 | 0x00000000 | TDFI\_PHYUPD\_TYPE3 and TDFI\_PHYUPD\_TYPE2 setting. |
| DENALI\_CTL\_127 | 0x1FC | 0x00000000 | Defines the DFI tPHYUPD\_RESP timing parameter |
| DENALI\_CTL\_128 | 0x200 | 0x00000000 | Defines the DFI tCTRLUPD\_INTERVAL timing parameter |
| DENALI\_CTL\_129 | 0x204 | 0x00020000 | TDFI\_DRAM\_CLK\_DISABLE  TDFI\_CTRL\_DELAY  WRLAT\_ADJ  RDLAT\_ADJ |
| DENALI\_CTL\_130 | 0x208 | 0x00000000 | TDFI\_WRLVL\_WW  TDFI\_WRLVL\_EN  TDFI\_DRAM\_CLK\_ENABLE |
| DENALI\_CTL\_131 | 0x20C | 0x00000000 | Defines the DFI tWRLVL\_RESP timing parameter |
| DENALI\_CTL\_132 | 0x210 | 0x00000000 | Defines the DFI tWRLVL\_MAX timing parameter |
| DENALI\_CTL\_133 | 0x214 | 0x00000000 | TDFI\_RDLVL\_RR  TDFI\_RDLVL\_EN |
| DENALI\_CTL\_134 | 0x218 | 0x00000000 | Defines the DFI tRDLVL\_RESP timing parameter |
| DENALI\_CTL\_135 | 0x21C | 0x00000000 | RDLVL\_GATE\_EN, RDLVL\_EN and RDLVL\_RESP\_MASK setting |
| DENALI\_CTL\_136 | 0x220 | 0x00000000 | Defines the DFI tRDLVL\_MAX timing parameter |
| DENALI\_CTL\_137 | 0x224 | 0x00000000 | RDLVL\_INTERVAL and RDLVL\_ERROR\_STATUS setting. |
| DENALI\_CTL\_138 | 0x228 | 0x00000000 | TDFI\_CALVL\_EN and RDLVL\_GATE\_INTERVAL setting. |
| DENALI\_CTL\_139 | 0x22C | 0x00000000 | TDFI\_CALVL\_CAPTURE and TDFI\_CALVL\_CC setting. |
| DENALI\_CTL\_140 | 0x230 | 0x00000000 | Defines the DFI tCALVL\_RESP timing parameter |
| DENALI\_CTL\_141 | 0x234 | 0x00000000 | Defines the DFI tCALVL\_MAX timing parameter |
| DENALI\_CTL\_142 | 0x238 | 0x00000000 | CALVL\_ERROR\_STATUS information. CALVL\_EN and CALVL\_RESP\_MASK setting. |
| DENALI\_CTL\_143 | 0x23C | 0x00000000 | TDFI\_RDCSLAT, TDFI\_PHY\_WRDATA and CALVL\_INTERVAL setting. |
| DENALI\_CTL\_144 | 0x240 | 0x00000000 | CTLR\_DISABLE\_ODT\_ON\_ZQ, EN\_1T\_TIMING and TDFI\_WRCSLAT setting. |
| **PHY Register** | | | |
| DENALI\_PHY\_00 | PHY\_BASE\_ADDR+0 | 0x00000000 | Controls PHY slice 0. |
| DENALI\_PHY\_01 | PHY\_BASE\_ADDR+1 | 0x00000000 | Controls PHY slice 0. |
| DENALI\_PHY\_02 | PHY\_BASE\_ADDR+2 | 0x00000000 | Controls PHY slice 0. |
| DENALI\_PHY\_03 | PHY\_BASE\_ADDR+3 | 0x00000000 | Controls PHY slice 0. |
| DENALI\_PHY\_04 | PHY\_BASE\_ADDR+4 | 0x00000000 | Controls the DLL for slice 0. |
| DENALI\_PHY\_05 | PHY\_BASE\_ADDR+5 | 0x00000000 | Controls the DLL for slice 0. |
| DENALI\_PHY\_06 | PHY\_BASE\_ADDR+6 | 0x00000000 | Controls PHY slice 0. |
| DENALI\_PHY\_07 | PHY\_BASE\_ADDR+7 | 0x00000000 | Controls loopback status, data and masking info for slice 0. |
| DENALI\_PHY\_08 | PHY\_BASE\_ADDR+8 | 0x00000000 | Reports DLL status for slice 0. |
| DENALI\_PHY\_09 | PHY\_BASE\_ADDR+9 | 0x00000000 | Reports DLL status for slice 0. |
| DENALI\_PHY\_10 | PHY\_BASE\_ADDR+10 | 0x00000000 | Reports DLL status for slice 0. |
| DENALI\_PHY\_11 | PHY\_BASE\_ADDR+11 | 0x00000000 | Debug command to continue leveling sequence 0. |
| DENALI\_PHY\_12 | PHY\_BASE\_ADDR+12 | 0x00000000 | Controls lvl timings for slice 0. |
| DENALI\_PHY\_13 | PHY\_BASE\_ADDR+13 | 0x00000000 | Reports GTLVL status for slice 0. |
| DENALI\_PHY\_14 | PHY\_BASE\_ADDR+14 | 0x00000000 | Reports RDLVL status for slice 0. |
| DENALI\_PHY\_15 | PHY\_BASE\_ADDR+15 | 0x00000000 | Controls wrlvl timings for slice 0. |
| DENALI\_PHY\_16 | PHY\_BASE\_ADDR+16 | 0x00000000 | Reports WRLVL status for slice 0. |
| DENALI\_PHY\_17 | PHY\_BASE\_ADDR+17 | - | This register intentionally blank. |
| DENALI\_PHY\_18 | PHY\_BASE\_ADDR+18 | - | This register intentionally blank. |
| DENALI\_PHY\_19 | PHY\_BASE\_ADDR+19 | - | This register intentionally blank. |
| DENALI\_PHY\_20 | PHY\_BASE\_ADDR+20 | - | This register intentionally blank. |
| DENALI\_PHY\_21 | PHY\_BASE\_ADDR+21 | - | This register intentionally blank. |
| DENALI\_PHY\_22 | PHY\_BASE\_ADDR+22 | - | This register intentionally blank. |
| DENALI\_PHY\_23 | PHY\_BASE\_ADDR+23 | - | This register intentionally blank. |
| DENALI\_PHY\_24 | PHY\_BASE\_ADDR+24 | - | This register intentionally blank. |
| DENALI\_PHY\_25 | PHY\_BASE\_ADDR+25 | - | This register intentionally blank. |
| DENALI\_PHY\_26 | PHY\_BASE\_ADDR+26 | - | This register intentionally blank. |
| DENALI\_PHY\_27 | PHY\_BASE\_ADDR+27 | - | This register intentionally blank. |
| DENALI\_PHY\_28 | PHY\_BASE\_ADDR+28 | - | This register intentionally blank. |
| DENALI\_PHY\_29 | PHY\_BASE\_ADDR+29 | - | This register intentionally blank. |
| DENALI\_PHY\_30 | PHY\_BASE\_ADDR+30 | - | This register intentionally blank. |
| DENALI\_PHY\_31 | PHY\_BASE\_ADDR+31 | - | This register intentionally blank. |
| DENALI\_PHY\_32 | PHY\_BASE\_ADDR+32 | 0x00000000 | Controls PHY slice 1. |
| DENALI\_PHY\_33 | PHY\_BASE\_ADDR+33 | 0x00000000 | Controls PHY slice 1. |
| DENALI\_PHY\_34 | PHY\_BASE\_ADDR+34 | 0x00000000 | Controls PHY slice 1. |
| DENALI\_PHY\_35 | PHY\_BASE\_ADDR+35 | 0x00000000 | Controls PHY slice 1. |
| DENALI\_PHY\_36 | PHY\_BASE\_ADDR+36 | 0x00000000 | Controls the DLL for slice 1. |
| DENALI\_PHY\_37 | PHY\_BASE\_ADDR+37 | 0x00000000 | Controls the DLL for slice 1. |
| DENALI\_PHY\_38 | PHY\_BASE\_ADDR+38 | 0x00000000 | Controls PHY slice 1. |
| DENALI\_PHY\_39 | PHY\_BASE\_ADDR+39 | 0x00000000 | Controls loopback status, data and masking info for slice 1. |
| DENALI\_PHY\_40 | PHY\_BASE\_ADDR+40 | 0x00000000 | Reports DLL status for slice 1. |
| DENALI\_PHY\_41 | PHY\_BASE\_ADDR+41 | 0x00000000 | Reports DLL status for slice 1. |
| DENALI\_PHY\_42 | PHY\_BASE\_ADDR+42 | 0x00000000 | Reports DLL status for slice 1. |
| DENALI\_PHY\_43 | PHY\_BASE\_ADDR+43 | 0x00000000 | Debug command to continue leveling sequence 1. |
| DENALI\_PHY\_44 | PHY\_BASE\_ADDR+44 | 0x00000000 | Controls lvl timings for slice 1. |
| DENALI\_PHY\_45 | PHY\_BASE\_ADDR+45 | 0x00000000 | Reports GTLVL status for slice 1. |
| DENALI\_PHY\_46 | PHY\_BASE\_ADDR+46 | 0x00000000 | Reports RDLVL status for slice 1. |
| DENALI\_PHY\_47 | PHY\_BASE\_ADDR+47 | 0x00000000 | Controls wrlvl timings for slice 1. |
| DENALI\_PHY\_48 | PHY\_BASE\_ADDR+48 | 0x00000000 | Reports WRLVL status for slice 1. |
| DENALI\_PHY\_49 | PHY\_BASE\_ADDR+49 | - | This register intentionally blank. |
| DENALI\_PHY\_50 | PHY\_BASE\_ADDR+50 | - | This register intentionally blank. |
| DENALI\_PHY\_51 | PHY\_BASE\_ADDR+51 | - | This register intentionally blank. |
| DENALI\_PHY\_52 | PHY\_BASE\_ADDR+52 | - | This register intentionally blank. |
| DENALI\_PHY\_53 | PHY\_BASE\_ADDR+53 | - | This register intentionally blank. |
| DENALI\_PHY\_54 | PHY\_BASE\_ADDR+54 | - | This register intentionally blank. |
| DENALI\_PHY\_55 | PHY\_BASE\_ADDR+55 | - | This register intentionally blank. |
| DENALI\_PHY\_56 | PHY\_BASE\_ADDR+56 | - | This register intentionally blank. |
| DENALI\_PHY\_57 | PHY\_BASE\_ADDR+57 | - | This register intentionally blank. |
| DENALI\_PHY\_58 | PHY\_BASE\_ADDR+58 | - | This register intentionally blank. |
| DENALI\_PHY\_59 | PHY\_BASE\_ADDR+59 | - | This register intentionally blank. |
| DENALI\_PHY\_60 | PHY\_BASE\_ADDR+60 | - | This register intentionally blank. |
| DENALI\_PHY\_61 | PHY\_BASE\_ADDR+61 | - | This register intentionally blank. |
| DENALI\_PHY\_62 | PHY\_BASE\_ADDR+62 | - | This register intentionally blank. |
| DENALI\_PHY\_63 | PHY\_BASE\_ADDR+63 | - | This register intentionally blank. |
| DENALI\_PHY\_64 | PHY\_BASE\_ADDR+64 | 0x00000000 | Controls PHY slice 2. |
| DENALI\_PHY\_65 | PHY\_BASE\_ADDR+65 | 0x00000000 | Controls PHY slice 2. |
| DENALI\_PHY\_66 | PHY\_BASE\_ADDR+66 | 0x00000000 | Controls PHY slice 2. |
| DENALI\_PHY\_67 | PHY\_BASE\_ADDR+67 | 0x00000000 | Controls PHY slice 2. |
| DENALI\_PHY\_68 | PHY\_BASE\_ADDR+68 | 0x00000000 | Controls the DLL for slice 2. |
| DENALI\_PHY\_69 | PHY\_BASE\_ADDR+69 | 0x00000000 | Controls the DLL for slice 2. |
| DENALI\_PHY\_70 | PHY\_BASE\_ADDR+70 | 0x00000000 | Controls PHY slice 2. |
| DENALI\_PHY\_71 | PHY\_BASE\_ADDR+71 | 0x00000000 | Controls loopback status, data and masking info for slice 2. |
| DENALI\_PHY\_72 | PHY\_BASE\_ADDR+72 | 0x00000000 | Reports DLL status for slice 2. |
| DENALI\_PHY\_73 | PHY\_BASE\_ADDR+73 | 0x00000000 | Reports DLL status for slice 2. |
| DENALI\_PHY\_74 | PHY\_BASE\_ADDR+74 | 0x00000000 | Reports DLL status for slice 2. |
| DENALI\_PHY\_75 | PHY\_BASE\_ADDR+75 | 0x00000000 | Debug command to continue leveling sequence 2. |
| DENALI\_PHY\_76 | PHY\_BASE\_ADDR+76 | 0x00000000 | Controls lvl timings for slice 2. |
| DENALI\_PHY\_77 | PHY\_BASE\_ADDR+77 | 0x00000000 | Reports GTLVL status for slice 2. |
| DENALI\_PHY\_78 | PHY\_BASE\_ADDR+78 | 0x00000000 | Reports RDLVL status for slice 2. |
| DENALI\_PHY\_79 | PHY\_BASE\_ADDR+79 | 0x00000000 | Controls wrlvl timings for slice 2. |
| DENALI\_PHY\_80 | PHY\_BASE\_ADDR+80 | 0x00000000 | Reports WRLVL status for slice 2. |
| DENALI\_PHY\_81 | PHY\_BASE\_ADDR+81 | - | This register intentionally blank. |
| DENALI\_PHY\_82 | PHY\_BASE\_ADDR+82 | - | This register intentionally blank. |
| DENALI\_PHY\_83 | PHY\_BASE\_ADDR+83 | - | This register intentionally blank. |
| DENALI\_PHY\_84 | PHY\_BASE\_ADDR+84 | - | This register intentionally blank. |
| DENALI\_PHY\_85 | PHY\_BASE\_ADDR+85 | - | This register intentionally blank. |
| DENALI\_PHY\_86 | PHY\_BASE\_ADDR+86 | - | This register intentionally blank. |
| DENALI\_PHY\_87 | PHY\_BASE\_ADDR+87 | - | This register intentionally blank. |
| DENALI\_PHY\_88 | PHY\_BASE\_ADDR+88 | - | This register intentionally blank. |
| DENALI\_PHY\_89 | PHY\_BASE\_ADDR+89 | - | This register intentionally blank. |
| DENALI\_PHY\_90 | PHY\_BASE\_ADDR+90 | - | This register intentionally blank. |
| DENALI\_PHY\_91 | PHY\_BASE\_ADDR+91 | - | This register intentionally blank. |
| DENALI\_PHY\_92 | PHY\_BASE\_ADDR+92 | - | This register intentionally blank. |
| DENALI\_PHY\_93 | PHY\_BASE\_ADDR+93 | - | This register intentionally blank. |
| DENALI\_PHY\_94 | PHY\_BASE\_ADDR+94 | - | This register intentionally blank. |
| DENALI\_PHY\_95 | PHY\_BASE\_ADDR+95 | - | This register intentionally blank. |
| DENALI\_PHY\_96 | PHY\_BASE\_ADDR+96 | 0x00000000 | Controls PHY slice 3. |
| DENALI\_PHY\_97 | PHY\_BASE\_ADDR+97 | 0x00000000 | Controls PHY slice 3. |
| DENALI\_PHY\_98 | PHY\_BASE\_ADDR+98 | 0x00000000 | Controls PHY slice 3. |
| DENALI\_PHY\_99 | PHY\_BASE\_ADDR+99 | 0x00000000 | Controls PHY slice 3. |
| DENALI\_PHY\_100 | PHY\_BASE\_ADDR+100 | 0x00000000 | Controls the DLL for slice 3. |
| DENALI\_PHY\_101 | PHY\_BASE\_ADDR+101 | 0x00000000 | Controls the DLL for slice 3. |
| DENALI\_PHY\_102 | PHY\_BASE\_ADDR+102 | 0x00000000 | Controls PHY slice 3. |
| DENALI\_PHY\_103 | PHY\_BASE\_ADDR+103 | 0x00000000 | Controls loopback status, data and masking info for slice 3. |
| DENALI\_PHY\_104 | PHY\_BASE\_ADDR+104 | 0x00000000 | Reports DLL status for slice 3. |
| DENALI\_PHY\_105 | PHY\_BASE\_ADDR+105 | 0x00000000 | Reports DLL status for slice 3. |
| DENALI\_PHY\_106 | PHY\_BASE\_ADDR+106 | 0x00000000 | Reports DLL status for slice 3. |
| DENALI\_PHY\_107 | PHY\_BASE\_ADDR+107 | 0x00000000 | Debug command to continue leveling sequence 3. |
| DENALI\_PHY\_108 | PHY\_BASE\_ADDR+108 | 0x00000000 | Controls lvl timings for slice 3. |
| DENALI\_PHY\_109 | PHY\_BASE\_ADDR+109 | 0x00000000 | Reports GTLVL status for slice 3. |
| DENALI\_PHY\_110 | PHY\_BASE\_ADDR+110 | 0x00000000 | Reports RDLVL status for slice 3. |
| DENALI\_PHY\_111 | PHY\_BASE\_ADDR+111 | 0x00000000 | Controls wrlvl timings for slice 3. |
| DENALI\_PHY\_112 | PHY\_BASE\_ADDR+112 | 0x00000000 | Reports WRLVL status for slice 3. |
| DENALI\_PHY\_113 | PHY\_BASE\_ADDR+113 | - | This register intentionally blank. |
| DENALI\_PHY\_114 | PHY\_BASE\_ADDR+114 | - | This register intentionally blank. |
| DENALI\_PHY\_115 | PHY\_BASE\_ADDR+115 | - | This register intentionally blank. |
| DENALI\_PHY\_116 | PHY\_BASE\_ADDR+116 | - | This register intentionally blank. |
| DENALI\_PHY\_117 | PHY\_BASE\_ADDR+117 | - | This register intentionally blank. |
| DENALI\_PHY\_118 | PHY\_BASE\_ADDR+118 | - | This register intentionally blank. |
| DENALI\_PHY\_119 | PHY\_BASE\_ADDR+119 | - | This register intentionally blank. |
| DENALI\_PHY\_120 | PHY\_BASE\_ADDR+120 | - | This register intentionally blank. |
| DENALI\_PHY\_121 | PHY\_BASE\_ADDR+121 | - | This register intentionally blank. |
| DENALI\_PHY\_122 | PHY\_BASE\_ADDR+122 | - | This register intentionally blank. |
| DENALI\_PHY\_123 | PHY\_BASE\_ADDR+123 | - | This register intentionally blank. |
| DENALI\_PHY\_124 | PHY\_BASE\_ADDR+124 | - | This register intentionally blank. |
| DENALI\_PHY\_125 | PHY\_BASE\_ADDR+125 | - | This register intentionally blank. |
| DENALI\_PHY\_126 | PHY\_BASE\_ADDR+126 | - | This register intentionally blank. |
| DENALI\_PHY\_127 | PHY\_BASE\_ADDR+127 | - | This register intentionally blank. |
| DENALI\_PHY\_128 | PHY\_BASE\_ADDR+128 | 0x00000000 | Controls PHY slice 4. |
| DENALI\_PHY\_129 | PHY\_BASE\_ADDR+129 | 0x00000000 | Controls PHY slice 4. |
| DENALI\_PHY\_130 | PHY\_BASE\_ADDR+130 | 0x00000000 | Controls PHY slice 4. |
| DENALI\_PHY\_131 | PHY\_BASE\_ADDR+131 | 0x00000000 | Controls PHY slice 4. |
| DENALI\_PHY\_132 | PHY\_BASE\_ADDR+132 | 0x00000000 | Controls the DLL for slice 4. |
| DENALI\_PHY\_133 | PHY\_BASE\_ADDR+133 | 0x00000000 | Controls the DLL for slice 4. |
| DENALI\_PHY\_134 | PHY\_BASE\_ADDR+134 | 0x00000000 | Controls PHY slice 4. |
| DENALI\_PHY\_135 | PHY\_BASE\_ADDR+135 | 0x00000000 | Controls loopback status, data and masking info for slice 4. |
| DENALI\_PHY\_136 | PHY\_BASE\_ADDR+136 | 0x00000000 | Reports DLL status for slice 4. |
| DENALI\_PHY\_137 | PHY\_BASE\_ADDR+137 | 0x00000000 | Reports DLL status for slice 4. |
| DENALI\_PHY\_138 | PHY\_BASE\_ADDR+138 | 0x00000000 | Reports DLL status for slice 4. |
| DENALI\_PHY\_139 | PHY\_BASE\_ADDR+139 | 0x00000000 | Debug command to continue leveling sequence 4. |
| DENALI\_PHY\_140 | PHY\_BASE\_ADDR+140 | 0x00000000 | Controls lvl timings for slice 4. |
| DENALI\_PHY\_141 | PHY\_BASE\_ADDR+141 | 0x00000000 | Reports GTLVL status for slice 4. |
| DENALI\_PHY\_142 | PHY\_BASE\_ADDR+142 | 0x00000000 | Reports RDLVL status for slice 4. |
| DENALI\_PHY\_143 | PHY\_BASE\_ADDR+143 | 0x00000000 | Controls wrlvl timings for slice 4. |
| DENALI\_PHY\_144 | PHY\_BASE\_ADDR+144 | 0x00000000 | Reports WRLVL status for slice 4. |
| DENALI\_PHY\_145 | PHY\_BASE\_ADDR+145 | - | This register intentionally blank. |
| DENALI\_PHY\_146 | PHY\_BASE\_ADDR+146 | - | This register intentionally blank. |
| DENALI\_PHY\_147 | PHY\_BASE\_ADDR+147 | - | This register intentionally blank. |
| DENALI\_PHY\_148 | PHY\_BASE\_ADDR+148 | - | This register intentionally blank. |
| DENALI\_PHY\_149 | PHY\_BASE\_ADDR+149 | - | This register intentionally blank. |
| DENALI\_PHY\_150 | PHY\_BASE\_ADDR+150 | - | This register intentionally blank. |
| DENALI\_PHY\_151 | PHY\_BASE\_ADDR+151 | - | This register intentionally blank. |
| DENALI\_PHY\_152 | PHY\_BASE\_ADDR+152 | - | This register intentionally blank. |
| DENALI\_PHY\_153 | PHY\_BASE\_ADDR+153 | - | This register intentionally blank. |
| DENALI\_PHY\_154 | PHY\_BASE\_ADDR+154 | - | This register intentionally blank. |
| DENALI\_PHY\_155 | PHY\_BASE\_ADDR+155 | - | This register intentionally blank. |
| DENALI\_PHY\_156 | PHY\_BASE\_ADDR+156 | - | This register intentionally blank. |
| DENALI\_PHY\_157 | PHY\_BASE\_ADDR+157 | - | This register intentionally blank. |
| DENALI\_PHY\_158 | PHY\_BASE\_ADDR+158 | - | This register intentionally blank. |
| DENALI\_PHY\_159 | PHY\_BASE\_ADDR+159 | - | This register intentionally blank. |
| DENALI\_PHY\_160 | PHY\_AC\_BASE\_ADDR+0 | 0x00000000 | Global controls for all slices |
| DENALI\_PHY\_161 | PHY\_AC\_BASE\_ADDR+1 | 0x00000000 | Reserve |
| DENALI\_PHY\_162 | PHY\_AC\_BASE\_ADDR+2 | 0x00000000 | CA leveling debug single step. |
| DENALI\_PHY\_163 | PHY\_AC\_BASE\_ADDR+3 | 0x00000000 | CA leveling controls for CA slice |
| DENALI\_PHY\_164 | PHY\_AC\_BASE\_ADDR+4 | 0x00000000 | CA leveling observe values. |
| DENALI\_PHY\_165 | PHY\_AC\_BASE\_ADDR+5 | 0x00000000 | Specifies the number of cycles that PHY takes to wakeup in Low Power Mode. |
| DENALI\_PHY\_166 | PHY\_AC\_BASE\_ADDR+6 | 0x00000000 | Global controls for all slices |
| DENALI\_PHY\_167 | PHY\_AC\_BASE\_ADDR+7 | 0x00000000 | Controls pad drive strengths |
| DENALI\_PHY\_168 | PHY\_AC\_BASE\_ADDR+8 | 0x00000000 | Controls pad drive strengths |
| DENALI\_PHY\_169 | PHY\_AC\_BASE\_ADDR+9 | 0x00000000 | Controls pad drive strengths |
| DENALI\_PHY\_170 | PHY\_AC\_BASE\_ADDR+10 | 0x00000000 | Controls pad termination values |
| DENALI\_PHY\_171 | PHY\_AC\_BASE\_ADDR+11 | 0x00000000 | Controls pad termination values |
| DENALI\_PHY\_172 | PHY\_AC\_BASE\_ADDR+12 | 0x00000000 | Controls calibration block values |
| DENALI\_PHY\_173 | PHY\_AC\_BASE\_ADDR+13 | 0x00000000 | Calibration block observation values. |
| DENALI\_PHY\_174 | PHY\_AC\_BASE\_ADDR+14 | 0x00000000 | Pad calibration interval counter compare value for block. |
|  |  |  | Reserve |
|  |  |  | Pad VREF SLP\_RE\_000\_1215V port settings. |

* 1. Register Descriptions
     1. Controller Register Descriptions

This chapter details the parameters included with this Cadence DDR Controller. Refer to the “Register Interface” Chapter for a description of the register interface and an overview of the register map, or the “Parameter Descriptions” Chapter for a description of every parameter in this controller.

DENALI\_CTL\_00

Address: Operational Base + offset (0x000)

Table .DENALI\_CTL\_00 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:21 | RD | 0x2041 | VERSION:  Holds the controller version number. |
| 20:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | DRAM\_CLASS:  Defines the class of DRAM memory which is connected to the controller. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | START:  Initiate command processing in the controller. Set to 1 to initiate. |

DENALI\_CTL\_01

Address: Operational Base + offset (0x004)

Table 538.DENALI\_CTL\_01 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RD | 0x00 | READ\_DATA\_FIFO\_DEPTH:  Reports the depth of the controller core read data queue. |
| 23:18 | - | - | Reserved |
| 17:16 | RD | 0x1 | MAX\_CS\_REG:  Holds the maximum number of chip selects available. |
| 15:12 | - | - | Reserved |
| 11:8 | RD | 0xc | MAX\_COL\_REG:  Holds the maximum width of column address in DRAMs. |
| 7:5 | - | - | Reserved |
| 4:0 | RD | 0xe | MAX\_ROW\_REG:  Holds the maximum width of memory address bus. |

DENALI\_CTL\_02

Address: Operational Base + offset (0x008)

Table 539.DENALI\_CTL\_02 register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RD | 0x00 | ASYNC\_CDC\_STAGES:  Reports the number of synchronizer delays specified for the asynchronous boundary crossings. |
| 23:16 | RD | 0x00 | WRITE\_DATA\_FIFO\_PTR\_WIDTH:  Reports the width of the controller core write data latency queue pointer. |
| 15:8 | RD | 0x00 | WRITE\_DATA\_FIFO\_DEPTH:  Reports the depth of the controller core write data latency queue. |
| 7:0 | RD | 0x00 | READ\_DATA\_FIFO\_PTR\_WIDTH:  Reports the width of the controller core read data queue pointer. |

DENALI\_CTL\_03

Address: Operational Base + offset (0x00C)

Table 540.DENALI\_CTL\_03register

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit** | **Type** | **Reset Value** | **Description** |
| 31:24 | RD | 0x00 | AXI0\_TRANS\_WRFIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 0 transition write data FIFO. Value is the log2 value of the depth. |
| 23:16 | RD | 0x00 | AXI0\_WR\_ARRAY\_LOG2\_DEPTH:  Reports the depth of the AXI port 0 write data array. Value is the log2 value of the depth. |
| 15:8 | RD | 0x00 | AXI0\_RDFIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 0 read data FIFO. Value is the log2 value of the depth. |
| 7:0 | RD | 0x00 | AXI0\_CMDFIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 0 command FIFO. Value is the log2 value of the depth. |

DENALI\_CTL\_04

Address: Operational Base + offset (0x010)

Table 541.DENALI\_CTL\_04egister

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RD | 0x00 | AXI1\_WR\_ARRAY\_LOG2\_DEPTH:  Reports the depth of the AXI port 1 write data array. Value is the log2 value of the depth. |
| 23:16 | RD | 0x00 | AXI1\_WR\_ARRAY\_LOG2\_DEPTH:  Reports the depth of the AXI port 1 read data FIFO. Value is the log2 value of the depth. |
| 15:8 | RD | 0x00 | AXI1\_CMDFIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 1 command FIFO. Value is the log2 value of the depth. |
| 7:0 | RD | 0x00 | AXI0\_WRCMD\_PROC\_FIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 0 write command processing FIFO. Value is the log2 value of the depth. |

DENALI\_CTL\_05

Address: Operational Base + offset (0x010)

Table 542.DENALI\_CTL\_05 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved |
| 15:8 | RD | 0x00 | AXI1\_WRCMD\_PROC\_FIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 1 write command processing FIFO. Value is the log2 value of the depth. |
| 7:0 | RD | 0x00 | AXI1\_TRANS\_WRFIFO\_LOG2\_DEPTH:  Reports the depth of the AXI port 1 transition write data FIFO. Value is  the log2 value of the depth. |

DENALI\_CTL\_06

Address: Operational Base + offset (0x014)

Table 543.DENALI\_CTL\_06 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_07

Address: Operational Base + offset (0x018)

Table 544.DENALI\_CTL\_07register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_08

Address: Operational Base + offset (0x020)

Table 545.DENALI\_CTL\_08 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:0 | RW | 0x000000 | TINIT: DRAM TINIT value in cycles. |

DENALI\_CTL\_09

Address: Operational Base + offset (0x024)

Table 546.DENALI\_CTL\_09register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | Reserved |
| 23:0 | RW | 0x000000 | TINIT3: DRAM TINIT3 value in cycles. |

DENALI\_CTL\_10

Address: Operational Base + offset (0x028)

Table 547.DENALI\_CTL\_10register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | Reserved |
| 23:0 | RW | 0x000000 | TINIT4: DRAM TINIT4 value in cycles. |

DENALI\_CTL\_11

Address: Operational Base + offset (0x02C)

Table 548.DENALI\_CTL\_11 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | NO\_AUTO\_MRR\_INIT: Disable MRR commands during initialization. Set to 1 to disable. |
| 23:0 | RW | 0x000000 | TINIT5: DRAM TINIT5 value in cycles. |

DENALI\_CTL\_12

Address: Operational Base + offset (0x030)

Table 549.DENALI\_CTL\_12register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_13

Address: Operational Base + offset (0x034)

Table 550.DENALI\_CTL\_13 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_14

Address: Operational Base + offset (0x038)

Table 551.DENALI\_CTL\_14register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:27 | - | - | Reserved |
| 26:24 | RW | 0x0 | TBST\_INT\_INTERVAL: DRAM burst interrupt interval value in cycles. |
| 23:16 | - | - | OBSOLETE |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | WRLAT: DRAM WRLAT value in cycles. |
| 7:6 | - | - | Reserved |
| 5:0 | RW | 0x00 | CASLAT\_LIN:  Sets latency from read command send to data receive from/to controller. Bit (0) is half-cycle increment and the upper bits define memory CAS latency for the controller. |

DENALI\_CTL\_15

Address: Operational Base + offset (0x03C)

Table 552.DENALI\_CTL\_15register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RW | 0x00 | TRAS\_MIN: DRAM TRAS\_MIN value in cycles. |
| 23:16 | RW | 0x00 | TRC: DRAM TRC value in cycles. |
| 15:8 | RW | 0x00 | TRRD: DRAM TRRD value in cycles. |
| 7:5 | - | - | Reserved |
| 4:0 | RW | 0x00 | TCCD: DRAM CAS-to-CAS value in cycles. |

DENALI\_CTL\_16

Address: Operational Base + offset (0x040)

Table 553.DENALI\_CTL\_16 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RW | 0x00 | TRTP:  DRAM TRTP value in cycles. |
| 23:16 | RW | 0x00 | TFAW:  DRAM TFAW value in cycles. |
| 15:8 | RW | 0x00 | TRP:  DRAM TRP value in cycles. |
| 7:6 | - | - | Reserved |
| 5:0 | RW | 0x00 | TWTR:  DRAM TWTR value in cycles. |

DENALI\_CTL\_17

Address: Operational Base + offset (0x044)

Table 554.DENALI\_CTL\_17 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:8 | RW | 0x00 | TMOD:  Number of cycles after MRS command and before any other command. |
| 7:0 | RW | 0x00 | TMRD:  DRAM TMRD value in cycles. |

DENALI\_CTL\_18

Address: Operational Base + offset (0x048)

Table 555.DENALI\_CTL\_18 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:24 | RW | 0x0 | TCKE: Minimum CKE pulse width. |
| 23:17 | - | - | Reserved |
| 16:0 | RW | 0x0000 | TRAS\_MAX: DRAM TRAS\_MAX value in cycles. |

DENALI\_CTL\_19

Address: Operational Base + offset (0x04C)

Table 556.DENALI\_CTL\_19 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:30 | - | - | Reserved |
| 29:24 | RW | 0x00 | TWR: DRAM TWR value in cycles. |
| 23:16 | RW | 0x00 | TRCD: DRAM TRCD value in cycles. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | WRITEINTERP:  Allow controller to interrupt a write burst to the DRAMs with a read command. Set to 1 to allow interruption. |
| 7:0 | RW | 0x00 | TCKESR: Minimum CKE low pulse width during a self-refresh. |

DENALI\_CTL\_20

Address: Operational Base + offset (0x050)

Table 557.DENALI\_CTL\_20 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:30 | - | - | Reserved |
| 29:24 | RW | 0x00 | TCAMRD: DRAM TCAMRD value in cycles. |
| 23:21 | - | - | Reserved |
| 20:16 | RW | 0x00 | TCAENT: DRAM TCAENT value in cycles. |
| 15:13 | - | - | Reserved |
| 12:8 | RW | 0x00 | TCACKEL: DRAM TCACKEL value in cycles. |
| 7:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | TMRR: DRAM TMRR value in cycles. |

DENALI\_CTL\_21

Address: Operational Base + offset (0x054)

Table 558.DENALI\_CTL\_21 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | AP: Enable auto pre-charge mode of controller. Set to 1 to enable. |
| 23:21 | - | - | Reserved |
| 20:16 | RW | 0x00 | TMRZ: DRAM TMRZ value in cycles. |
| 15:13 | - | - | Reserved |
| 12:8 | RW | 0x00 | TCACKEH: DRAM TCACKEH value in cycles. |
| 7:5 | - | - | Reserved |
| 4:0 | RW | 0x00 | TCAEXT: DRAM TCAEXT value in cycles. |

DENALI\_CTL\_22

Address: Operational Base + offset (0x058)

Table 559.DENALI\_CTL\_22 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:27 | - | - | Reserved |
| 26:24 | RW\_D | 0x2 | BSTLEN: Encoded burst length sent to DRAMs during initialization. Program to 1 for BL2, program to 2 for BL4, or program to 3 for BL8. |
| 23:16 | RW | 0x00 | TDAL: DRAM TDAL value in cycles |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | TRAS\_LOCKOUT: IF the DRAM supports it, this allows the controller to execute auto pre-charge commands before the TRAS\_MIN parameter expires. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | CONCURRENTAP: IF the DRAM supports it, this allows the controller to issue commands to other banks while a bank is in auto pre-charge. Set to 1 to enable. |

DENALI\_CTL\_23

Address: Operational Base + offset (0x05C)

Table 560.DENALI\_CTL\_23 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | WR | 0x0 | AREFRESH: Initiate auto-refresh at the end of the current burst boundary. Set to 1 to trigger. |
| 23:16 | - | - | OBSOLETE |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | REG\_DIMM\_ENABLE: Enable registered DIMM operation of the controller. Set to 1 to enable. |
| 7:0 | RW | 0x00 | TRP\_AB: DRAM TRP all bank value in cycles. |

DENALI\_CTL\_24

Address: Operational Base + offset (0x060)

Table 561.DENALI\_CTL\_24 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:16 | RW | 0x000 | TRFC: DRAM TRFC value in cycles. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | TREF\_ENABLE: Issue auto-refresh commands to the DRAMs at the interval defined in the TREF parameter. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | RESERVED: Reserved for future use. Refer to the regconfig files for the default programming. |

DENALI\_CTL\_25

Address: Operational Base + offset (0x064)

Table 562.DENALI\_CTL\_25 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:0 | RW | 0x0000 | TREF: DRAM TREF value in cycles. |

DENALI\_CTL\_26

Address: Operational Base + offset (0x068)

Table 563.DENALI\_CTL\_26 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | Reserved |
| 23:8 | RW | 0x000000 | TPDEX: DRAM TPDEX value in cycles. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_27

Address: Operational Base + offset (0x06C)

Table 564.DENALI\_CTL\_27 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:16 | RW | 0x00 | TMRRI: DRAM TMRRI value in cycles. |
| 15:0 | - | - | OBSOLETE |

DENALI\_CTL\_28

Address: Operational Base + offset (0x070)

Table 565.DENALI\_CTL\_28 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | RW | 0x0000 | TXSNR: DRAM TXSNR value in cycles. |
| 15:0 | RW | 0x0000 | TXSR: DRAM TXSR value in cycles. |

DENALI\_CTL\_29

Address: Operational Base + offset (0x074)

Table 566.DENALI\_CTL\_29 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:27 | - | - | Reserved |
| 26:24 | RW | 0x0 | CKE\_DELAY: Additional cycles to delay CKE for status reporting. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | ENABLE\_QUICK\_SREFRESH: Allow user to interrupt memory initialization to enter self-refresh mode. Set to 1 to allow interruption. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | RESERVED: Reserved for future use. Refer to the regconfig files for the default  programming. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | PWRUP\_SREFRESH\_EXIT: Allow power up via self-refresh instead of full memory initialization. Set to 1 to enable. |

DENALI\_CTL\_30

Address: Operational Base + offset (0x078)

Table 567.DENALI\_CTL\_30 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | WR | 0x00 | LP\_CMD: Low power software command request interface. Bit (0) controls exit, bit (1) controls entry, bits (4:2) define the low power state, bit (5) controls memory clock gating, bit (6) controls controller clock gating, and bit (7) controls lock. |
| 23:16 | RW | 0x00 | CKSRX: Clock stable delay on self-refresh exit. |
| 15:8 | RW | 0x00 | CKSRE: Clock hold delay on self-refresh entry. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | LOWPOWER\_REFRESH\_ENABLE: Enable refreshes while in low power mode. Bit (0) controls cs0, bit (1) controls cs1, etc. Set each bit to 1 to disable. |

DENALI\_CTL\_31

Address: Operational Base + offset (0x07C)

Table 568.DENALI\_CTL\_31 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:24 | RW | 0x0 | LPI\_TIMER\_WAKEUP: Defines the DFI tLP\_WAKEUP timing parameter (in DFI clocks) to be driven when the LPI timer expires. |
| 23:20 | - | - | Reserved |
| 19:16 | RW | 0x0 | LPI\_SR\_MCCLK\_GATE\_WAKEUP: Defines the DFI tLP\_WAKEUP timing parameter (in DFI clocks) to be driven when memory is in self-refresh with memory and controller clock gating. |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | LPI\_SR\_WAKEUP: Defines the DFI tLP\_WAKEUP timing parameter (in DFI clocks) to be driven when memory is in self-refresh. |
| 7:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | LPI\_PD\_WAKEUP: Defines the DFI tLP\_WAKEUP timing parameter (in DFI clocks) to be driven when memory is in power-down. |

DENALI\_CTL\_32

Address: Operational Base + offset (0x080)

Table 569.DENALI\_CTL\_32 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:8 | RW | 0x000 | LPI\_TIMER\_COUNT: Defines the LPI timer count. |
| 7:5 | - | - | Reserved |
| 4:0 | RW | 0x00 | LPI\_WAKEUP\_EN: Enables the various low power wakeup parameters. Bit (0) enables power-down wakeup, bit (1) enables self-refresh wakeup, bit (2) enables self-refresh with memory and controller clock gating wakeup, bit (3) is reserved and bit (4) enables the LPI timer. Set each bit to 1 to enable. |

DENALI\_CTL\_33

Address: Operational Base + offset (0x084)

Table 570.DENALI\_CTL\_33 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:30 | - | - | Reserved |
| 29:24 | RD | 0x20 | LP\_STATE: Low power state status parameter. Bits (4:0) indicate the current low power state and bit (5) set indicates that status bits are valid. |
| 23:19 | - | - | Reserved |
| 18:16 | RW | 0x0 | TDFI\_LP\_RESP: Defines the DFI tLP\_RESP timing parameter (in DFI clocks), the maximum cycles between a dfi\_lp\_req assertion and a dfi\_lp\_ack assertion. |
| 15:12 | - | - | Reserved |
| 11:0 | RW | 0x000 | LPI\_WAKEUP\_TIMEOUT: Defines the LPI timeout time, the maximum cycles between a dfi\_lp\_req de-assertion and a dfi\_lp\_ack de-assertion. If this value is exceeded, an interrupt will occur. |

DENALI\_CTL\_34

Address: Operational Base + offset (0x088)

Table 571.DENALI\_CTL\_34 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:24 | RW | 0x0 | LP\_AUTO\_MEM\_GATE\_EN: Enable memory clock gating when entering a low power state via the auto low power counters. Bit (0) controls power-down, and bit (1) controls self-refresh. Set each bit to 1 to enable. |
| 23:19 | - | - | Reserved |
| 18:16 | RW | 0x0 | LP\_AUTO\_EXIT\_EN: Enable auto exit from each of the low power states when a read or write command enters the command queue. Bit (0) controls power-down, bit (1) controls self-refresh and bit (2) controls self-refresh with memory and controller clock gating. Set each bit to 1 to enable. |
| 15:11 | - | - | Reserved |
| 10:8 | RW | 0x0 | LP\_AUTO\_ENTRY\_EN: Enable auto entry into each of the low power states when the associated idle timer expires. Bit (0) controls power-down, bit (1) controls self-refresh, and bit (2) controls self-refresh with memory and controller clock gating. Set each bit to 1 to enable. |
| 7:4 | - | - | Reserved |
| 3:0 | RD | 0x0 | LP\_ARB\_STATE: Reports on the state of the arbiter. Bits (2:0) indicate which interface has control of the low power control module and bit (3) indicates if the software programmable interface has an active lock on the arbiter. For bits (2:0), value of 0 indicates module is idle, value of 1 indicates software programmable interface is in control, value of 2 indicates external pin interface is in control, value of 3 indicates automatic interface is in control, value of 4 indicates dynamic power control per chip select interface is in control, and value of 5 indicates that the controller is in control. |

DENALI\_CTL\_35

Address: Operational Base + offset (0x08C)

Table 572.DENALI\_CTL\_35 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RW | 0x00 | LP\_AUTO\_SR\_MC\_GATE\_IDLE: Number of long count sequences until the controller will place memory in self-refresh with controller and memory clock gating. |
| 23:16 | RW | 0x00 | LP\_AUTO\_SR\_IDLE: Number of long count sequences until the controller will place memory in self-refresh. |
| 15:12 | - | - | Reserved |
| 11:0 | RW | 0x000 | LP\_AUTO\_PD\_IDLE: Defines the idle time until the controller will place memory in active power-down. |

DENALI\_CTL\_36

Address: Operational Base + offset (0x090)

Table 573.DENALI\_CTL\_36 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | OBSOLETE |
| 7 | - | - | Reserved |
| 6:0 | RW | 0x00 | RESERVED: Reserved for future use. Refer to the regconfig files for the default programming. |

DENALI\_CTL\_37

Address: Operational Base + offset (0x094)

Table 574.DENALI\_CTL\_37 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:27 | - | - | Reserved |
| 26:0 | RW+ | 0x0000000 | WRITE\_MODEREG: Write memory mode register data to the DRAMs. Bits (7:0) define the memory mode register number if bit (23) is set, bits (15:8) define the chip select if bit (24) is clear, bits (23:16) define which memory mode register/s to write, bit (24) defines whether all chip selects will be written, and bit (25) triggers the write. |

DENALI\_CTL\_38

Address: Operational Base + offset (0x098)

Table 575.DENALI\_CTL\_38 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24:8 | RW+ | 0x00000 | READ\_MODEREG: Read the specified memory mode register from specified chip when start bit set. Bits (7:0) define the memory mode register and bits (15:8) define the chip select. Set bit (16) to 1 to trigger. |
| 7:0 | RD | 0x00 | MRW\_STATUS: Write memory mode register status. Bit (0) set indicates a WRITE\_MODEREG parameter programming error. Bit (1) set indicates a PASR error. Bit (2) is Reserved. Bit (3) set indicates a self refresh or deep power down error. Bit (4) is Reserved. |

DENALI\_CTL\_39

Address: Operational Base + offset (0x09C)

Table 576.DENALI\_CTL\_39 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | PERIPHERAL\_MRR\_DATA [31:0]: Data and chip returned from memory mode register read requested by the READ\_MODEREG parameter. Bits (7:0) indicate the read data and bits (15:8) indicate the chip. |

DENALI\_CTL\_40

Address: Operational Base + offset (0x0A0)

Table 577.DENALI\_CTL\_40 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:8 | RD | 0x0000 | AUTO\_TEMPCHK\_VAL\_0: MR4 data for all devices on chip 0 accessed by automatic MRR commands. Bits (3:0) correlate to the device on the lower byte, bits (7:4) correlate to the devices on the 2nd byte etc. Value indicates the OP7, OP2, OP1 and OP0 bits. |
| 7:0 | RD | 0x00 | PERIPHERAL\_MRR\_DATA [39:32]:  Data and chip returned from memory mode register read requested by the READ\_MODEREG parameter. Bits (7:0) indicate the read data and bits (15:8) indicate the chip. |

DENALI\_CTL\_41

Address: Operational Base + offset (0x0A4)

Table 578.DENALI\_CTL\_41 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | RW | 0x0000 | REFRESH\_PER\_AUTO\_TEMPCHK:  Number of long count sequences between automatic memory mode register read commands of MR4. |
| 15:0 | - | - | OBSOLETE |

DENALI\_CTL\_42

Address: Operational Base + offset (0x0A8)

Table 579.DENALI\_CTL\_42 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:16 | RW | 0x0000 | MR1\_DATA\_0: Data to program into memory mode register 1 for chip select 0. |
| 15:0 | - | - | OBSOLETE |

DENALI\_CTL\_43

Address: Operational Base + offset (0x0AC)

Table 580.DENALI\_CTL\_43 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:16 | RW | 0x0000 | MRSINGLE\_DATA\_0: Data to program into memory mode register single write to chip select 0. |
| 15:8 | - | - | OBSOLETE |
| 7:0 | RW | 0x0000 | MR2\_DATA\_0: Data to program into memory mode register 2 for chip select 0. |

DENALI\_CTL\_44

Address: Operational Base + offset (0x0B0)

Table 581.DENALI\_CTL\_44 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RW | 0x00 | MR11\_DATA\_0: Data to program into memory mode register 11 for chip select 0 |
| 23:16 | RD | 0x00 | MR8\_DATA\_0: Data read from MR8 for chip select 0. |
| 15:8 | - | - | OBSOLETE |
| 7:0 | RW | 0x0000 | MR3\_DATA\_0: Data to program into memory mode register 3 for chip select 0 |

DENALI\_CTL\_45

Address: Operational Base + offset (0x0B4)

Table 582.DENALI\_CTL\_45 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:8 | RW | 0x00 | MR17\_DATA\_0: Data to program into memory mode register 17 for chip select 0 |
| 7:0 | RW | 0x00 | MR16\_DATA\_0: Data to program into memory mode register 16 for chip select 0 |

DENALI\_CTL\_46

Address: Operational Base + offset (0x0B8)

Table 583.DENALI\_CTL\_46 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_47

Address: Operational Base + offset (0x0BC)

Table 584.DENALI\_CTL\_47 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_48

Address: Operational Base + offset (0x0C0)

Table 585.DENALI\_CTL\_48 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_49

Address: Operational Base + offset (0x0C4)

Table 586.DENALI\_CTL\_49 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:29 | - | - | Reserved |
| 28:24 | RW | 0x00 | ADDR\_SPACE:  Sets the number of address bits to check during BIST operation. |
| 23:18 | - | - | Reserved |
| 17:16 | RD | 0x0 | BIST\_RESULT:  BIST operation status (pass/fail). Bit (0) indicates data check status and bit (1) indicates address check status. Value of 1 is a passing result. |
| 15:9 | - | - | Reserved |
| 8 | WR | 0x0 | BIST\_GO:  Initiate a BIST operation. Set to 1 to trigger. |
| 7:1 | - | - | Reserved |
| 0 | RD | 0x0 | RL3\_SUPPORT\_EN:  Indicates if RL3 is supported by a connected LPDDR3 memory. Data read from MR0 bit 7. |

DENALI\_CTL\_50

Address: Operational Base + offset (0x0C8)

Table 587.DENALI\_CTL\_50 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | BIST\_ADDR\_CHECK:  Enable address checking with BIST operation. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | BIST\_DATA\_CHECK:  Enable data checking with BIST operation. Set to 1 to enable. |

DENALI\_CTL\_51

Address: Operational Base + offset (0x0CC)

Table 588.DENALI\_CTL\_51 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | - | - | OBSOLETE |
| 30:0 | RW | 0x00000000 | BIST\_START\_ADDRESS :  Start BIST checking at this address. |

DENALI\_CTL\_52

Address: Operational Base + offset (0x0D0)

Table 589.DENALI\_CTL\_52 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_53

Address: Operational Base + offset (0x0D4)

Table 590.DENALI\_CTL\_53 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | BIST\_DATA\_MASK [31:0]:  Mask applied to data for BIST error checking. Bit (0) controls memory data path bit (0), bit (1) controls memory data path bit (1), etc. Set each bit to 1 to mask. |

DENALI\_CTL\_54

Address: Operational Base + offset (0x0D8)

Table 591.DENALI\_CTL\_54 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | BIST\_DATA\_MASK [63:32]:  Mask applied to data for BIST error checking. Bit (0) controls memory data path bit (0), bit (1) controls memory data path bit (1), etc. Set each bit to 1 to mask. |

DENALI\_CTL\_55

Address: Operational Base + offset (0x0DC)

Table 592.DENALI\_CTL\_55 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:8 | RW\_D | 0x000 | ZQINIT:  Number of cycles needed for a ZQINIT command. |
| 7:5 | - | - | Reserved |
| 4:0 | RW | 0x00 | LONG\_COUNT\_MASK:  Reduces the length of the long counter from 1024 cycles. The only supported values are 0x00 (1024 cycles), 0x10 (512 clocks), 0x18 (256 clocks), 0x1C (128 clocks), 0x1E (64 clocks) and 0x1F (32 clocks). |

DENALI\_CTL\_56

Address: Operational Base + offset (0x0E0)

Table 593.DENALI\_CTL\_56 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:16 | RW | 0x000 | ZQCS: Number of cycles needed for a ZQCS command. |
| 15:12 | - | - | Reserved |
| 11:0 | RW | 0x000 | ZQCL: Number of cycles needed for a ZQCL command. |

DENALI\_CTL\_57

Address: Operational Base + offset (0x0E4)

Table 594.DENALI\_CTL\_57 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | ZQ\_ON\_SREF\_EXIT: Defines the type of ZQ calibration performed at self-refresh exit. Program to 0x1 for ZQ Short (ZQCS), program to 0x2 for ZQ Long (ZQCL), program to 0x4 for ZQ Initialization (ZQINIT), or program to 0x8 for ZQ Reset. Clearing to 0x0 will not trigger any ZQ command on self-refresh exit. |
| 7:4 | - | - | Reserved |
| 3:0 | WR | 0x0 | ZQ\_REQ: User request to initiate a ZQ calibration. Program to 0x1 for ZQ Short (ZQCS), program to 0x2 for ZQ Long (ZQCL), program to 0x4 for ZQ Initialization (ZQINIT), or program to 0x8 for ZQ Reset. Clearing to 0x0 will not trigger any ZQ command. This parameter should only be written when the ZQ\_IN\_PROGRESS parameter is cleared to 0. |

DENALI\_CTL\_58

Address: Operational Base + offset (0x0E8)

Table 595.DENALI\_CTL\_58 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | ZQ\_INTERVAL: Number of long count sequences allowed between automatic ZQCS commands. |

DENALI\_CTL\_59

Address: Operational Base + offset (0x0EC)

Table 596.DENALI\_CTL\_59 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | NO\_ZQ\_INIT: Disable ZQ operations during initialization. Set to 1 to disable. |
| 23:20 | - | - | Reserved |
| 19:8 | RW | 0x000 | ZQRESET: Number of cycles needed for a ZQRESET command. |
| 7:1 | - | - | Reserved |
| 0 | RD | 0x0 | ZQ\_IN\_PROGRESS: Indicates that a ZQ command is currently running. Value of 1 indicates command in progress. |

DENALI\_CTL\_60

Address: Operational Base + offset (0x0F0)

Table 597.DENALI\_CTL\_60 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:24 | RW | 0x0 | COL\_DIFF:  Difference between number of column pins available and number being used. |
| 23:19 | - | - | Reserved |
| 18:16 | RW | 0x0 | ROW\_DIFF:  Difference between number of address pins available and number being used. |
| 15:10 | - | - | Reserved |
| 9:8 | RW | 0x0 | BANK\_DIFF:  Encoded number of banks on the DRAM(s). |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_61

Address: Operational Base + offset (0x0F4)

Table 598.DENALI\_CTL\_61 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | RW | 0x00 | AGE\_COUNT:  Initial value of master aging-rate counter for command aging. |
| 23:20 | - | - | Reserved |
| 19:16 | RW\_D | 0xa | APREBIT:  Location of the auto pre-charge bit in the DRAM address. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | BANK\_ADDR\_INTLV\_EN:  Enables the capability to interleave the bank address within the row address bits. Set to 1 to enable. |
| 7:5 | - | - | Reserved |
| 4:0 | RW | 0x00 | BANK\_START\_BIT:  Defines the LSbit of the bank address within the page of the user address when the BANK\_ADDR\_INTLV\_EN parameter is set. |

DENALI\_CTL\_62

Address: Operational Base + offset (0x0F8)

Table 599.DENALI\_CTL\_62 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | BANK\_SPLIT\_EN:  Enable bank splitting as a rule for command queue placement. Set to 1 to enable. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | RESERVED:  Reserved for future use. Refer to the regconfig files for the default programming. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | ADDR\_CMP\_EN:  Enable address collision detection as a rule for command queue placement. Set to 1 to enable. |
| 7:0 | RW | 0x00 | COMMAND\_AGE\_COUNT:  Initial value of individual command aging counters for command aging. |

DENALI\_CTL\_63

Address: Operational Base + offset (0x0FC)

Table 600.DENALI\_CTL\_63 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | RW\_SAME\_PAGE\_EN:  Enable page grouping when read/ write grouping as a rule for command queue placement. This is only valid when the RW\_SAME\_EN parameter is set. Set to 1 to enable. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | RW\_SAME\_EN:  Enable read/write grouping as a rule for command queue placement. Set to 1 to enable. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | PRIORITY\_EN:  Enable priority as a rule for command queue placement. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | PLACEMENT\_EN:  Enable placement logic for command queue. Set to 1 to enable. |

DENALI\_CTL\_64

Address: Operational Base + offset (0x100)

Table 601.DENALI\_CTL\_64 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:24 | RW | 0x0 | NUM\_Q\_ENTRIES\_ACT\_DISABLE:  Number of queue entries in which ACT requests will be disabled. Setting to X will disable ACT requests from the X entries lowest in the command queue. |
| 23:18 | - | - | Reserved |
| 17:16 | RW | 0x0 | DISABLE\_RW\_GROUP\_W\_BNK\_CONFLIC T:  Disables placement to read/write group when grouping creates a bank collision. Bit (0) controls placement next to bank conflict command and bit (1) controls placement 2 away from bank conflict command. Set each bit to 1 to disable. |
| 15:0 | - | - | Reserved |

DENALI\_CTL\_65

Address: Operational Base + offset (0x104)

Table 602.DENALI\_CTL\_65 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:18 | - | - | Reserved |
| 17:16 | RW | 0x0 | INHIBIT\_DRAM\_CMD:  Inhibit command types from being executed from the command queue. Clear to 0 to enable any command, program to 1 to inhibit read/write and bank commands, program to 2 to inhibit MRR and peripheral MRR commands, or program to 3 to inhibit MRR and read/write commands. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | DISABLE\_RD\_INTERLEAVE:  Disable read data interleaving for commands from the same port, regardless of the requestor ID. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | SWAP\_EN:  Enable command swapping logic in execution unit. Set to 1 to enable. |

DENALI\_CTL\_66

Address: Operational Base + offset (0x108)

Table 603.DENALI\_CTL\_66 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:27 | - | - | Reserved |
| 26:24 | RW | 0x0 | MEMDATA\_RATIO\_0:  Defines the ratio of the DRAM device size on chip select 0 to the memory data width. Program with the log2 ratio of the memory data width to the device data width. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | LPDDR2\_S4:  Defines the DRAM LPDDR2 device type being used. Set to 1 for LPDDR2-S4. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | REDUC:  Enable the half datapath feature of the controller. Set to 1 to enable. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_67

Address: Operational Base + offset (0x10C)

Table 604.DENALI\_CTL\_67 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:24 | RW | 0x0 | WR\_ORDER\_REQ:  Determines if the controller can re-order write commands from the same source ID and/or the same port. Bit (0) controls source ID usage and bit (1) controls port ID usage. Set each bit to 1 to enable usage in placement logic. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | IN\_ORDER\_ACCEPT:  Forces the controller to accept commands in the order in which they are placed in the command queue. |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | Q\_FULLNESS:  Quantity that determines command queue full. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_68

Address: Operational Base + offset (0x110)

Table 605.DENALI\_CTL\_68 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:29 | - | - | Reserved |
| 28:24 | RD | 0x00 | DFI\_ERROR:  Indicates that the DFI error flag has been asserted. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | CTRLUPD\_REQ\_PER\_AREF\_EN:  Enable an automatic controller-initiated update (dfi\_ctrlupd\_req) after every refresh. Set to 1 to enable. |
| 15:9 | - | - | Reserved |
| 8 | WR | 0x0 | CTRLUPD\_REQ:  Assert the DFI controller-initiated update request signal dfi\_ctrlupd\_req. Set to 1 to trigger. |
| 7:1 | - | - | Reserved |
| 0 | RD | 0x0 | CONTROLLER\_BUSY:  Indicator that the controller is processing a command. Evaluates all ports for outstanding transactions. Value of 1 indicates controller busy. |

DENALI\_CTL\_69

Address: Operational Base + offset (0x114)

Table 606.DENALI\_CTL\_69 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RD | 0x00000 | DFI\_ERROR\_INFO:  Holds the encoded DFI error type associated with the DFI\_ERROR parameter assertion. |

DENALI\_CTL\_70

Address: Operational Base + offset (0x118)

Table 607.DENALI\_CTL\_70 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:0 | RD | 0x0000000 | INT\_STATUS:  Status of interrupt features in the controller. |

DENALI\_CTL\_71

Address: Operational Base + offset (0x11C)

Table 608.DENALI\_CTL\_71 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24:0 | WR | 0x0000000 | INT\_ACK:  Clear mask of the INT\_STATUS parameter. |

DENALI\_CTL\_72

Address: Operational Base + offset (0x120)

Table 609.DENALI\_CTL\_72 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:0 | RW | 0x0000000 | INT\_MASK :  Mask for the controller\_int signal from the INT\_STATUS parameter |

DENALI\_CTL\_73

Address: Operational Base + offset (0x124)

Table 610.DENALI\_CTL\_73 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | - | - | Reserved |
| 30:0 | RD | 0x00000000 | OUT\_OF\_RANGE\_ADDR:  Address of command that caused an out-of-range interrupt. |

DENALI\_CTL\_74

Address: Operational Base + offset (0x128)

Table 611.DENALI\_CTL\_74 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23 | - | - | Reserved |
| 22:16 | RD | 0x00 | OUT\_OF\_RANGE\_TYPE:  Type of command that caused an out-of-range interrupt. |
| 15 | - | - | Reserved |
| 14:8 | RD | 0x00 | OUT\_OF\_RANGE\_LENGTH:  Length of command that caused an out-of-range interrupt. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_75

Address: Operational Base + offset (0x12C)

Table 612.DENALI\_CTL\_75 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:9 | - | - | Reserved |
| 8:0 | RD | 0x000 | OUT\_OF\_RANGE\_SOURCE\_ID:  Source ID of command that caused an out-of-range interrupt. |

DENALI\_CTL\_76

Address: Operational Base + offset (0x130)

Table 613.DENALI\_CTL\_76 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_EXP\_DATA [31:0]: Expected data on BIST error. |

DENALI\_CTL\_77

Address: Operational Base + offset (0x134)

Table 614.DENALI\_CTL\_77 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_EXP\_DATA [63:32]: Expected data on BIST error. |

DENALI\_CTL\_78

Address: Operational Base + offset (0x138)

Table 615.DENALI\_CTL\_78 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_EXP\_DATA [95:64]: Expected data on BIST error. |

DENALI\_CTL\_79

Address: Operational Base + offset (0x13C)

Table 616.DENALI\_CTL\_79 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_EXP\_DATA [127:96]: Expected data on BIST error. |

DENALI\_CTL\_80

Address: Operational Base + offset (0x140)

Table 617.DENALI\_CTL\_80 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_FAIL\_DATA [31:0]: Actual data on BIST error. |

DENALI\_CTL\_81

Address: Operational Base + offset (0x144)

Table 618.DENALI\_CTL\_81 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_FAIL\_DATA [63:32]: Actual data on BIST error. |

DENALI\_CTL\_82

Address: Operational Base + offset (0x148)

Table 619.DENALI\_CTL\_82 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_FAIL\_DATA [95:64]: Actual data on BIST error. |

DENALI\_CTL\_83

Address: Operational Base + offset (0x14C)

Table 620.DENALI\_CTL\_83 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RD | 0x00000000 | BIST\_FAIL\_DATA [127:96]: Actual data on BIST error. |

DENALI\_CTL\_84

Address: Operational Base + offset (0x150)

Table 621.DENALI\_CTL\_84 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | - | - | Reserved |
| 30:0 | RD | 0x00000000 | BIST\_FAIL\_ADDR : Address of BIST error. |

DENALI\_CTL\_85

Address: Operational Base + offset (0x154)

Table 622.DENALI\_CTL\_85 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_86

Address: Operational Base + offset (0x158)

Table 623.DENALI\_CTL\_86 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | - | - | Reserved |
| 31:0 | RD | 0x00000000 | PORT\_CMD\_ERROR\_ADDR : Address of command that caused the PORT command error. |

DENALI\_CTL\_87

Address: Operational Base + offset (0x15C)

Table 624.DENALI\_CTL\_87 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:24 | RD | 0x0 | PORT\_CMD\_ERROR\_TYPE: Type of error and access type that caused the PORT command error. |
| 23:17 | - | - | Reserved |
| 16:8 | RD | 0x000 | PORT\_CMD\_ERROR\_ID: Source ID of command that caused the PORT command error. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_88

Address: Operational Base + offset (0x160)

Table 625.DENALI\_CTL\_88 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_89

Address: Operational Base + offset (0x164)

Table 626.DENALI\_CTL\_89 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | ODT\_EN:  Enable support of DRAM ODT. When enabled, controller will assert and de-assert ODT output to DRAM as needed. |
| 23:16 | - | - | OBSOLETE |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | TODTH\_WR:  Defines the DRAM minimum ODT high time after an ODT assertion for a write command. |
| 7:0 | RW | 0x00 | TODTL\_2CMD:  Defines the DRAM delay from an ODT de-assertion to the next non-write, non-read command. |

DENALI\_CTL\_90

Address: Operational Base + offset (0x168)

Table 627.DENALI\_CTL\_90 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:13 | - | - | Reserved |
| 12:8 | RW | 0x00 | RESERVED: Reserved for future use. Refer to the regconfig files for the default  Programming. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | EN\_ODT\_ASSERT\_EXCEPT\_RD: Enable controller to assert ODT at all times except during reads. Assumes single ODT pin connected. Set to 1 to enable. |

DENALI\_CTL\_91

Address: Operational Base + offset (0x16C)

Table 628.DENALI\_CTL\_91 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:29 | - | - | Reserved |
| 28:24 | RW | 0x00 | R2R\_SAMECS\_DLY: Additional delay to insert between two reads to the same chip select. Any value including 0x0 supported. |
| 23:0 | - | - | OBSOLETE |

DENALI\_CTL\_92

Address: Operational Base + offset (0x170)

Table 629.DENALI\_CTL\_92 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:24 | RW | 0x0 | TDQSCK\_MAX: Additional delay needed for tDQSCK. |
| 23:21 | - | - | Reserved |
| 20:16 | RW | 0x0 | W2W\_SAMECS\_DLY: Additional delay to insert between two writes to the same chip select. Any value including 0x0 supported. |
| 15:13 | - | - | Reserved |
| 12:8 | RW | 0x0 | W2R\_SAMECS\_DLY: Additional delay to insert between writes and reads to the same chip select. |
| 7:5 | - | - | Reserved |
| 4:0 | RW\_D | 0x02 | R2W\_SAMECS\_DLY: Additional delay to insert between reads and writes to the same chip select. Program to a non-zero value. |

DENALI\_CTL\_93

Address: Operational Base + offset (0x174)

Table 630.DENALI\_CTL\_93 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | WR | 0x0 | SWLVL\_START: User request to initiate software leveling of type in the SW\_LEVELING\_MODE parameter. Set to 1 to trigger. |
| 23:17 | - | - | Reserved |
| 16 | WR | 0x0 | SWLVL\_LOAD: User request to load delays and execute software leveling. Set to 1 to trigger. |
| 15:11 | - | - | Reserved |
| 10:8 | RW | 0x0 | SW\_LEVELING\_MODE: Defines the leveling operation for software leveling. Clear to 0 for none, program to 1 for write leveling, program to 2 for data eye training, or program to 3 for gate training. |
| 7:2 | - | - | OBSOLETE |

DENALI\_CTL\_94

Address: Operational Base + offset (0x178)

Table 631.DENALI\_CTL\_94 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RD | 0x0 | SWLVL\_RESP\_1: Leveling response for data slice 1. |
| 23:17 | - | - | Reserved |
| 16 | RD | 0x0 | SWLVL\_RESP\_0: Leveling response for data slice 0. |
| 15:9 | - | - | Reserved |
| 8 | RD | 0x0 | SWLVL\_OP\_DONE: Signals that software leveling is currently in progress. Value of 1 indicates operation complete. |
| 7:1 | - | - | Reserved |
| 0 | WR | 0x0 | SWLVL\_EXIT: User request to exit software leveling. Set to 1 to exit. |

DENALI\_CTL\_95

Address: Operational Base + offset (0x17C)

Table 632.DENALI\_CTL\_95 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | WRLVL\_CS: Specifies the target chip select for the write leveling operation initiated through the WRLVL\_REQ parameter. |
| 23:17 | - | - | Reserved |
| 16 | WR | 0x0 | WRLVL\_REQ: User request to initiate write leveling. Set to 1 to trigger. |
| 15:9 | - | - | Reserved |
| 8 | RD | 0x0 | SWLVL\_RESP\_3: Leveling response for data slice 3. |
| 7:1 | - | - | Reserved |
| 0 | RD | 0x0 | SWLVL\_RESP\_2: Leveling response for data slice 2. |

DENALI\_CTL\_96

Address: Operational Base + offset (0x180)

Table 633.DENALI\_CTL\_96 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:17 | - | - | Reserved |
| 16 | RW | 0x0 | WRLVL\_EN: Enable the MC write leveling module. Set to 1 to enable. |
| 15:14 | - | - | Reserved |
| 13:8 | RW | 0x00 | WLMRD: Delay from issuing MRS to first write leveling strobe. |
| 7:6 | - | - | Reserved |
| 5:0 | RW | 0x00 | WLDQSEN: Delay from issuing MRS to first DQS strobe for write leveling. |

DENALI\_CTL\_97

Address: Operational Base + offset (0x184)

Table 634.DENALI\_CTL\_97 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | WRLVL\_PERIODIC: Enables the use of the dfi\_lvl\_periodic signal during write leveling. Set to 1 to enable. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | DFI\_PHY\_WRLVL\_MODE: Specifies the PHY support for DFI write leveling. Set to 1 for supported. |
| 15:0 | RW | 0x0000 | WRLVL\_INTERVAL: Number of long count sequences counted between automatic write leveling commands. |

DENALI\_CTL\_98

Address: Operational Base + offset (0x188)

Table 635.DENALI\_CTL\_98 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | WRLVL\_AREF\_EN: Enables refreshes and other non-data commands to execute in the middle of write leveling. Set to 1 to enable. |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | WRLVL\_RESP\_MASK: Mask for the dfi\_wrlvl\_resp signal during write leveling. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | WRLVL\_ON\_SREF\_EXIT:  Enables automatic write leveling on a self-refresh exit. Set to 1 to enable. |

DENALI\_CTL\_99

Address: Operational Base + offset (0x18C)

Table 636.DENALI\_CTL\_99 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | RDLVL\_GATE\_REQ: User request to initiate gate training. Set to 1 to trigger. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | RDLVL\_REQ: User request to initiate data eye training. Set to 1 to trigger. |
| 15:10 | - | - | Reserved |
| 9:8 | RD | 0x0 | WRLVL\_ERROR\_STATUS: Holds the error associated with the write level error interrupt. Bit (0) set indicates a TDFI\_WRLVL\_MAX parameter violation and bit (1) set indicates a TDFI\_WRLVL\_RESP parameter violation. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_100

Address: Operational Base + offset (0x190)

Table 637.DENALI\_CTL\_100 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | DFI\_PHY\_RDLVL\_MODE: Specifies the PHY support for DFI data eye training. Set to 1 for supported. |
| 23:18 | - | - | Reserved |
| 17:16 | RW | 0x0 | RDLVL\_GATE\_SEQ\_EN: Specifies the pattern, format and MPR for gate training. |
| 15:10 | - | - | Reserved |
| 9:8 | RW | 0x0 | RDLVL\_SEQ\_EN: Specifies the pattern, format and MPR for data eye training. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | RDLVL\_CS: Specifies the target chip select for the data eye training operation initiated through the RDLVL\_REQ parameter or the gate training operation initiated through the RDLVL\_GATE\_REQ parameter. |

DENALI\_CTL\_101

Address: Operational Base + offset (0x194)

Table 638.DENALI\_CTL\_101 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | RDLVL\_GATE\_PERIODIC: Enables the use of the dfi\_lvl\_periodic signal during gate training. Set to 1 to enable. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | RDLVL\_ON\_SREF\_EXIT: Enables automatic data eye training on a self-refresh exit. Set to 1 to enable. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | RDLVL\_PERIODIC: Enables the use of the dfi\_lvl\_periodic signal during data eye training. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | DFI\_PHY\_RDLVL\_GATE\_MODE: Specifies the PHY support for DFI gate training. Set to 1 for supported. |

DENALI\_CTL\_102

Address: Operational Base + offset (0x198)

Table 639.DENALI\_CTL\_102 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | RESERVED: Reserved for future use. Refer to the regconfig files for the default programming. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | RDLVL\_AREF\_EN: Enables refreshes and other non-data commands to execute in the middle of data eye training or gate training. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | RDLVL\_GATE\_ON\_SREF\_EXIT: Enables automatic gate training on a self-refresh exit. Set to 1 to enable. |

DENALI\_CTL\_103

Address: Operational Base + offset (0x19C)

Table 640.DENALI\_CTL\_103 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | WR | 0x0 | CALVL\_REQ: User request to initiate CA training. Set to 1 to trigger. |
| 23:0 | - | - | OBSOLETE |

DENALI\_CTL\_104

Address: Operational Base + offset (0x1A0)

Table 641.DENALI\_CTL\_104 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:8 | RW | 0x00000 | CALVL\_PAT\_0: CA Training pattern 0 driven on the CA bus during a calibration command. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | CALVL\_CS: Specifies the target chip select for the CA training operation initiated through the CALVL\_REQ parameter. |

DENALI\_CTL\_105

Address: Operational Base + offset (0x1A4)

Table 642.DENALI\_CTL\_105 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_BG\_PAT\_0: CA Training pattern 0 driven on the CA bus before and after a calibration command. |

DENALI\_CTL\_106

Address: Operational Base + offset (0x1A8)

Table 643.DENALI\_CTL\_106 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_PAT\_1: CA Training pattern 1 driven on the CA bus during a calibration command. |

DENALI\_CTL\_107

Address: Operational Base + offset (0x1AC)

Table 644.DENALI\_CTL\_107 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_BG\_PAT\_1: CA Training pattern 1 driven on the CA bus before and after a calibration command. |

DENALI\_CTL\_108

Address: Operational Base + offset (0x1B0)

DENALI\_CTL\_108 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_PAT\_2: CA Training pattern 2 driven on the CA bus during a calibration command. |

DENALI\_CTL\_109

Address: Operational Base + offset (0x1B4)

Table 645.DENALI\_CTL\_109 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_BG\_PAT\_2: CA Training pattern 2 driven on the CA bus before and after a calibration command. |

DENALI\_CTL\_110

Address: Operational Base + offset (0x1B8)

Table 646.DENALI\_CTL\_110 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_PAT\_3: CA Training pattern 3 driven on the CA bus during a calibration command. |

DENALI\_CTL\_111

Address: Operational Base + offset (0x1BC)

Table 647.DENALI\_CTL\_111 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | RESERVED: Reserved for future use. Refer to the regconfig files for the default programming. |
| 23:20 | - | - | Reserved |
| 19:0 | RW | 0x00000 | CALVL\_BG\_PAT\_3: CA Training pattern 3 driven on the CA bus before and after a calibration command. |

DENALI\_CTL\_112

Address: Operational Base + offset (0x1C0)

Table 648.DENALI\_CTL\_112 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | Reserved |
| 24 | RW | 0x0 | CALVL\_PERIODIC: Enables the use of the dfi\_lvl\_periodic signal during CA training. Set to 1 to enable. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | DFI\_PHY\_CALVL\_MODE: Specifies the PHY support for DFI CA training. Set to 1 for supported. |
| 15:10 | - | - | Reserved |
| 9:8 | RW | 0x0 | CALVL\_SEQ\_EN: Specifies which CA training patterns will be used. Clear to 0 for pattern 0 only, program to 1 for patterns 0 and 1, program to 2 for patterns 0, 1 and 2, or program to 3 for all patterns. |
| 7:4 | - | - | Reserved |
| 3:0 | - | - | RESERVED: Reserved for future use. Refer to the regconfig files for the default programming. |

DENALI\_CTL\_113

Address: Operational Base + offset (0x1C4)

Table 649.DENALI\_CTL\_113 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | OBSOLETE |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | CALVL\_AREF\_EN: Enables refreshes and other non-data commands to execute in the middle of CA training. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | CALVL\_ON\_SREF\_EXIT: Enables automatic CA training on a self-refresh exit. Set to 1 to enable. |

DENALI\_CTL\_114

Address: Operational Base + offset (0x1C8)

Table 650.DENALI\_CTL\_114 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:24 | RW | 0x0 | AXI0\_FIFO\_TYPE\_REG: Clock domain relativity between AXI port 0 and the controller core. Clear to 0 for asynchronous, program to 1 for 2:1 port:core pseudo-sync, program to 2 for 1:2 port:core pseudo-sync, or program to 3 for synchronous. |
| 23:20 | - | - | Reserved |
| 19:16 | RW | 0x0 | AXI0\_W\_PRIORITY: Priority of write commands from AXI port 0. 0 is the highest priority. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and the AXI0\_FIXED\_PORT\_PRIORITY\_ENABLE parameter is low. |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | AXI0\_R\_PRIORITY: Priority of read commands from AXI port 0. 0 is the highest priority. This may only be changed before initialization begins or when the  controller is quiescent, there is no data in the port FIFOs, and the AXI0\_FIXED\_PORT\_PRIORITY\_ENABLE parameter is low. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | AXI0\_FIXED\_PORT\_PRIORITY\_ENABLE: Defines the priority control for AXI port 0 as per-port or per-command. Set to 1 for per-port with priority defined through the AXI.8.\_R\_PRIORITY and AXI.8.\_W\_PRIORITY parameters. Clear to 0 for per-command. |

DENALI\_CTL\_115

Address: Operational Base + offset (0x1CC)

Table 651.DENALI\_CTL\_115 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:24 | RW | 0x0 | AXI1\_FIFO\_TYPE\_REG: Clock domain relativity between AXI port 1 and the controller core. Clear to 0 for asynchronous,program to 1 for 2:1 port:core pseudo-sync, program to 2 for 1:2 port:core pseudo-sync, or program to 3 for synchronous. |
| 23:20 | - | - | Reserved |
| 19:16 | RW | 0x0 | AXI1\_W\_PRIORITY: Priority of write commands from AXI port 1. 0 is the highest priority. This may only be changed before initialization begins or when the  controller is quiescent, there is no data in the port FIFOs, and the  AXI1\_FIXED\_PORT\_PRIORITY\_ENABLE parameter is low. |
| 15:12 | - | - | Reserved |
| 11:8 | RW | 0x0 | AXI1\_R\_PRIORITY: Priority of read commands from AXI port 1. 0 is the highest priority. This may only be changed before initialization begins or when the  controller is quiescent, there is no data in the port FIFOs, and the AXI1\_FIXED\_PORT\_PRIORITY\_ENABLE parameter is low. |
| 7:1 | - | - | Reserved |
| 0 | RW | 0x0 | AXI1\_FIXED\_PORT\_PRIORITY\_ENABLE: Defines the priority control for AXI port 1 as per-port or per-command. Set to 1 for per-port with priority defined through the AXI.8.\_R\_PRIORITY and AXI.8.\_W\_PRIORITY parameters. Clear to 0 for per-command. |

DENALI\_CTL\_116

Address: Operational Base + offset (0x1D0)

Table 652.DENALI\_CTL\_116 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | - | - | OBSOLETE |

DENALI\_CTL\_117

Address: Operational Base + offset (0x1D4)

Table 653.DENALI\_CTL\_117 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | OBSOLETE |

DENALI\_CTL\_118

Address: Operational Base + offset (0x1D8)

Table 654.DENALI\_CTL\_118 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31 | - | - | Reserved |
| 30:24 | RD | 0x00 | AXI0\_CURRENT\_BDW: Current bandwidth usage percentage for port 0. |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | AXI0\_BDW\_OVFLOW: Port 0 behavior when bandwidth maximized. Set to 1 to allow overflow. |
| 15 | - | - | Reserved |
| 14:8 | RW | 0x00 | AXI0\_BDW: Maximum bandwidth percentage for port 0. |
| 7:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | ARB\_CMD\_Q\_THRESHOLD: Threshold for command queue fullness related to overflow. |

DENALI\_CTL\_119

Address: Operational Base + offset (0x1DC)

Table 655.DENALI\_CTL\_119 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23 | - | - | Reserved |
| 22:16 | RD | 0x00 | AXI1\_CURRENT\_BDW: Current bandwidth usage percentage for port 1. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | AXI1\_BDW\_OVFLOW: Port 1 behavior when bandwidth maximized. Set to 1 to allow overflow. |
| 7 | - | - | Reserved |
| 6:0 | RW | 0x00 | AXI1\_BDW: Maximum bandwidth percentage for port 1. |

DENALI\_CTL\_120

Address: Operational Base + offset (0x1E0)

Table 656.DENALI\_CTL\_120 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:25 | - | - | OBSOLETE |

DENALI\_CTL\_121

Address: Operational Base + offset (0x1E4)

Table 657.DENALI\_CTL\_121 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:9 | - | - | Reserved |
| 9:8 | RD | 0x0 | CKE\_STATUS: Register access to cke\_status signal. |
| 7:0 | - | - | OBSOLETE |

DENALI\_CTL\_122

Address: Operational Base + offset (0x1E8)

Table 658.DENALI\_CTL\_122 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:30 | - | - | Reserved |
| 29:24 | RD | Calc Value | TDFI\_PHY\_WRLAT: Holds the calculated DFI tPHY\_WRLAT timing parameter (in DFI PHY clocks), the maximum cycles between a write command and a dfi\_wrdata\_en assertion. |
| 23:16 | RW | 0x00 | DLL\_RST\_ADJ\_DLY: Minimum cycles after setting master delay in DLL until the DLL reset signal dll\_rst\_n may be asserted. If this signal is not being used by the PHY, this parameter may be ignored. |
| 15:0 | RW | 0x0000 | DLL\_RST\_DELAY: Minimum cycles required for DLL reset signal dll\_rst\_n to be held. If this signal is not being used by the PHY, this parameter may be ignored. |

DENALI\_CTL\_123

Address: Operational Base + offset (0x1EC)

Table 659.DENALI\_CTL\_123 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:24 | RW | 0x0 | DRAM\_CLK\_DISABLE: Set value for the dfi\_dram\_clk\_disable signal. Bit (0) controls cs0, bit (1) controls cs1, etc. Set each bit to 1 to disable. |
| 23:22 | - | - | Reserved |
| 21:16 | RD | Calc Value | TDFI\_RDDATA\_EN: Holds the calculated DFI tRDDATA\_EN timing parameter (in DFI PHY clocks), the maximum cycles between a read command and a dfi\_rddata\_en assertion. |
| 15:14 | - | - | Reserved |
| 13:8 | RW\_D | 0x06 | TDFI\_PHY\_RDLAT: Defines the DFI tPHY\_RDLATtiming parameter (in DFI PHYclocks), the maximum cycles between a dfi\_rddata\_en assertion and a dfi\_rddata\_valid assertion. |
| 7 | - | - | Reserved |
| 6:0 | RD | 0x00 | UPDATE\_ERROR\_STATUS: Identifies the source of any DFI MC-initiated or PHY-initiated update errors. Value of 1 indicates a timing violation of the associated timing parameter. |

DENALI\_CTL\_124

Address: Operational Base + offset (0x1F0)

Table 660.DENALI\_CTL\_124 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:8 | RW | 0x0000 | TDFI\_CTRLUPD\_MAX: Defines the DFI tCTRLUPD\_MAX timing parameter (in DFI clocks), the maximum cycles that dfi\_ctrlupd\_req can be asserted. If programmed to a non-zero, a timing violation will cause an interrupt and bit (1) set in the UPDATE\_ERROR\_STATUS parameter. |
| 7:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | TDFI\_CTRLUPD\_MIN: Reports the DFI tCTRLUPD\_MIN timing parameter (in DFI clocks), the minimum cycles that dfi\_ctrlupd\_req must be asserted. |

DENALI\_CTL\_125

Address: Operational Base + offset (0x1F4)

Table 661.DENALI\_CTL\_125 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | RW | 0x0000 | TDFI\_PHYUPD\_TYPE1: Defines the DFI tPHYUPD\_TYPE1 timing parameter (in DFI clocks), the maximum cycles that dfi\_phyupd\_req can assert after dfi\_phyupd\_ack for dfi\_phyupd\_type 1. If programmed to a non-zero, a timing violation will cause an interrupt and bit (3) set in the UPDATE\_ERROR\_STATUS parameter. |
| 15:0 | RW | 0x0000 | TDFI\_PHYUPD\_TYPE0: Defines the DFI tPHYUPD\_TYPE0 timing parameter (in DFI clocks), the maximum cycles that dfi\_phyupd\_req can assert after dfi\_phyupd\_ack for dfi\_phyupd\_type 0. If programmed to a non-zero, a timing violation will cause an interrupt and bit (2) set in the UPDATE\_ERROR\_STATUS parameter. |

DENALI\_CTL\_126

Address: Operational Base + offset (0x1F8)

Table 662.DENALI\_CTL\_126 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | RW | 0x0000 | TDFI\_PHYUPD\_TYPE3: Defines the DFI tPHYUPD\_TYPE3 timing parameter (in DFI clocks), the maximum cycles that dfi\_phyupd\_req can assert after dfi\_phyupd\_ack for dfi\_phyupd\_type 3. If programmed to a non-zero, a timing violation will cause an interrupt and bit (5) set in the UPDATE\_ERROR\_STATUS parameter. |
| 15:0 | RW | 0x0000 | TDFI\_PHYUPD\_TYPE2: Defines the DFI tPHYUPD\_TYPE2 timing parameter (in DFI clocks), the maximum cycles that dfi\_phyupd\_req can assert after dfi\_phyupd\_ack for dfi\_phyupd\_type 2. If programmed to a non-zero, a timing violation will cause an interrupt and bit (4) set in the UPDATE\_ERROR\_STATUS parameter. |

DENALI\_CTL\_127

Address: Operational Base + offset (0x1FC)

Table 663.DENALI\_CTL\_127 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | OBSOLETE |
| 15:0 | RW | 0x0000 | TDFI\_PHYUPD\_RESP: Defines the DFI tPHYUPD\_RESP timing parameter (in DFI clocks), the maximum cycles between a dfi\_phyupd\_req assertion and a dfi\_phyupd\_ack assertion. If programmed to a non-zero, a timing violation will cause an interrupt and bit (6) set in the UPDATE\_ERROR\_STATUS parameter. |

DENALI\_CTL\_128

Address: Operational Base + offset (0x200)

Table 664.DENALI\_CTL\_128 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x000000  00 | TDFI\_CTRLUPD\_INTERVAL: Defines the DFI tCTRLUPD\_INTERVAL timing parameter (in DFI clocks), the maximum cycles between dfi\_ctrlupd\_req assertions. If programmed to a non-zero, a timing violation will cause an interrupt and bit (0) set in the UPDATE\_ERROR\_STATUS parameter. |

DENALI\_CTL\_129

Address: Operational Base + offset (0x204)

Table 665.DENALI\_CTL\_129 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:28 | - | - | Reserved |
| 27:24 | RW | 0x0 | TDFI\_DRAM\_CLK\_DISABLE: Defines the DFI tDRAM\_CLK\_DISABLE timing parameter (in DFI clocks), the delay between a dfi\_dram\_clock\_disable assertion and the memory clock disable. |
| 23:20 | - | - | Reserved |
| 19:16 | RW\_D | 0x2 | TDFI\_CTRL\_DELAY: Defines the DFI tCTRL\_DELAY timing parameter (in DFI clocks), the delay between a DFI command change and a memory command. |
| 15:14 | - | - | Reserved |
| 13:8 | RW | 0x00 | WRLAT\_ADJ: Adjustment value for PHY write timing. |
| 7:6 | - | - | Reserved |
| 5:0 | RW | 0x00 | RDLAT\_ADJ: Adjustment value for PHY read timing. |

DENALI\_CTL\_130

Address: Operational Base + offset (0x208)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:16 | RW | 0x000 | TDFI\_WRLVL\_WW: Defines the DFI tWRLVL\_WW timing parameter (in DFI clocks), the minimum cycles between dfi\_wrlvl\_strobe assertions. |
| 15:8 | RW | 0x00 | TDFI\_WRLVL\_EN: Defines the DFI tWRLVL\_EN timing parameter (in DFI clocks), the minimum cycles from a dfi\_wrlvl\_en assertion to the first dfi\_wrlvl\_strobe assertion. |
| 7:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | TDFI\_DRAM\_CLK\_ENABLE: Defines the DFI tDRAM\_CLK\_ENABLE timing parameter (in DFI clocks), the delay between a dfi\_dram\_clk\_disable de-assertion and the memory clock enable. |

DENALI\_CTL\_131

Address: Operational Base + offset (0x20C)

Table 666.DENALI\_CTL\_131 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | TDFI\_WRLVL\_RESP: Defines the DFI tWRLVL\_RESP timing parameter (in DFI clocks), the maximum cycles between a dfi\_wrlvl\_req assertion and a dfi\_wrlvl\_en assertion. |

DENALI\_CTL\_132

Address: Operational Base + offset (0x210)

Table 667.DENALI\_CTL\_132 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | TDFI\_WRLVL\_MAX: Defines the DFI tWRLVL\_MAX timing parameter (in DFI clocks), the maximum cycles between a dfi\_wrlvl\_en assertion and a valid dfi\_wrlvl\_resp. |

DENALI\_CTL\_133

Address: Operational Base + offset (0x214)

Table 668.DENALI\_CTL\_133 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:18 | - | - | Reserved |
| 17:8 | RW | 0x000 | TDFI\_RDLVL\_RR: Defines the DFI tRDLVL\_RR timing parameter (in DFI clocks), the minimum cycles between read commands. |
| 7:0 | RW | 0x00 | TDFI\_RDLVL\_EN: Defines the DFI tRDLVL\_EN timing parameter (in DFI clocks), the minimum cycles from a dfi\_rdlvl\_en or dfi\_rdlvl\_gate\_en assertion to the first read or MRR. |

DENALI\_CTL\_134

Address: Operational Base + offset (0x218)

Table 669.DENALI\_CTL\_134 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | TDFI\_RDLVL\_RESP: Defines the DFI tRDLVL\_RESP timing parameter (in DFI clocks), the maximum cycles between a dfi\_rdlvl\_req or dfi\_rdlvl\_gate\_req assertion and a dfi\_rdlvl\_en or dfi\_rdlvl\_gate\_en assertion. |

DENALI\_CTL\_135

Address: Operational Base + offset (0x21C)

Table 670.DENALI\_CTL\_135 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | RDLVL\_GATE\_EN:Enable the MC gate training module. Set to 1 to enable. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | RDLVL\_EN: Enable the MC data eye training module. Set to 1 to enable. |
| 7:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | RDLVL\_RESP\_MASK: Mask for the dfi\_rdlvl\_resp signal during data eye training. |

DENALI\_CTL\_136

Address: Operational Base + offset (0x220)

Table 671.DENALI\_CTL\_136 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | TDFI\_RDLVL\_MAX: Defines the DFI tRDLVL\_MAX timing parameter (in DFI clocks), the maximum cycles between a dfi\_rdlvl\_en or dfi\_rdlvl\_gate\_en assertion and a valid dfi\_rdlvl\_resp. |

DENALI\_CTL\_137

Address: Operational Base + offset (0x224)

Table 672.DENALI\_CTL\_137 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:8 | RW | 0x0000 | RDLVL\_INTERVAL: Number of long count sequences counted between automatic data eye training commands. |
| 7:2 | - | - | Reserved |
| 1:0 | RD | 0x0 | RDLVL\_ERROR\_STATUS: Holds the error associated with the data eye training error or gate training error interrupt. Uppermost bit set indicates a TDFI\_RDLVL\_RESP parameter violation. Next uppermost bit set indicates a TDFI\_RDLVL\_MAX parameter violation. Lower bits are reserved. |

DENALI\_CTL\_138

Address: Operational Base + offset (0x228)

Table 673.DENALI\_CTL\_138 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:16 | RW | 0x00 | TDFI\_CALVL\_EN: Defines the DFI tCALVL\_EN timing parameter (in DFI clocks), the minimum cycles between a dfi\_calvl\_en assertion and a dfi\_cke de-assertion. |
| 15:0 | RW | 0x0000 | RDLVL\_GATE\_INTERVAL: Number of long count sequences counted between automatic gate training commands. |

DENALI\_CTL\_139

Address: Operational Base + offset (0x22C)

Table 674.DENALI\_CTL\_139 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:26 | - | - | Reserved |
| 25:16 | RW | 0x000 | TDFI\_CALVL\_CAPTURE: Defines the DFI tCALVL\_CAPTURE timing parameter (in DFI clocks), the minimum cycles between a calibration command and a dfi\_calvl\_capture pulse. |
| 15:10 | - | - | Reserved |
| 9:0 | RW | 0x000 | TDFI\_CALVL\_CC: Defines the DFI tCALVL\_CC timing parameter (in DFI clocks), the minimum cycles between calibration commands. |

DENALI\_CTL\_140

Address: Operational Base + offset (0x230)

Table 675.DENALI\_CTL\_140 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x00000000 | TDFI\_CALVL\_RESP: Defines the DFI tCALVL\_RESP timing parameter (in DFI clocks), the maximum cycles between a dfi\_calvl\_req assertion and a dfi\_calvl\_en assertion. |

DENALI\_CTL\_141

Address: Operational Base + offset (0x234)

Table 676.DENALI\_CTL\_141 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0x000000  00 | TDFI\_CALVL\_MAX: Defines the DFI tCALVL\_MAX timing parameter (in DFI clocks), the maximum cycles between a dfi\_calvl\_en assertion and a valid dfi\_calvl\_resp. |

DENALI\_CTL\_142

Address: Operational Base + offset (0x238)

Table 677.DENALI\_CTL\_142 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:18 | - | - | Reserved |
| 17:16 | RD | 0x0 | CALVL\_ERROR\_STATUS: Holds the error associated with the CA training error interrupt. Bit (0) set indicates a TDFI\_CALVL\_RESP parameter violation and bit (1) set indicates a TDFI\_CALVL\_MAX parameter violation. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | CALVL\_EN: Enable the MC CA training module. Set to 1 to enable. |
| 7:1 | - | - | Reserved |
| 0 |  | 0x0 | CALVL\_RESP\_MASK: Mask for the dfi\_calvl\_resp signal during CA training. |

DENALI\_CTL\_143

Address: Operational Base + offset (0x23C)

Table 678.DENALI\_CTL\_143 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:30 | - | - | Reserved |
| 29:24 | RW | 0x00 | TDFI\_RDCSLAT: Defines the DFI tPHY\_RDCSLAT timing parameter (in DFI PHY clocks), the maximum cycles between a read command and a dfi\_rddata\_cs\_n assertion. |
| 23:19 | - | - | Reserved |
| 18:16 | RW | 0x0 | TDFI\_PHY\_WRDATA: Defines the DFI tPHY\_WRDATA timing parameter (in DFI PHY clocks), the maximum cycles between a dfi\_wrdata\_en assertion and a dfi\_wrdata signal. |
| 15:0 | RW | 0x0000 | CALVL\_INTERVAL: Number of long count sequences counted between automatic CA training commands. |

DENALI\_CTL\_144

Address: Operational Base + offset (0x240)

Table 679.DENALI\_CTL\_144 register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | - | - | OBSOLETE |
| 23:17 | - | - | Reserved |
| 16 | RW | 0x0 | CTLR\_DISABLE\_ODT\_ON\_ZQ: Forces an automatic MRW to disable ODT before ZQ command is issued. Set to 1 for ODT disable. |
| 15:9 | - | - | Reserved |
| 8 | RW | 0x0 | EN\_1T\_TIMING: Enable 1T timing in a system supporting both 1T and 2T timing. Set to 1 to enable. |
| 7:6 | - | - | Reserved |
| 5:0 | RW | 0x00 | TDFI\_WRCSLAT: Defines the DFI tPHY\_WRCSLAT timing parameter (in DFI PHY clocks), the maximum cycles between a write command and a dfi\_wrdata\_cs\_n assertion. |

* + 1. PHY Register Descriptions

This chapter details the parameters included with this Cadence Low-Power DDR PHY. Refer to the “Register Interface” Chapter for a description of the register interface and an overview of the register map.

Table . Program Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Bit | Attr | Reset Value | Description |
| DENALI\_PHY\_00 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQ\_TIMING\_REG\_0: Controls PHY slice 0. |
| DENALI\_PHY\_01 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQS\_TIMING\_REG\_0: Controls PHY slice 0. |
| DENALI\_PHY\_02 | [31:0] | RW | 0x00000000 | DEN\_PHY\_GATE\_LPBK\_CTRL\_REG\_0:Controls PHY slice 0. |
| DENALI\_PHY\_03 | [31:0] | RW | 0x00000000 | DEN\_PHY\_READ\_CTRL\_REG\_0: Controls PHY slice 0. |
| DENALI\_PHY\_04 | [31:0] | RW | 0x00000000 | PHY\_DLL\_MASTER\_CTRL\_REG\_0: Controls the DLL for slice 0. |
| DENALI\_PHY\_05 | [31:0] | RW+ | 0x00000000 | PHY\_DLL\_SLAVE\_CTRL\_REG\_0: Controls the DLL for slice 0. |
| DENALI\_PHY\_06 | [31:0] | RW | 0x00000000 | DEN\_PHY\_IE\_TIMING\_REG\_0: Controls PHY slice 0. |
| DENALI\_PHY\_07 | [31:0] | RD | 0x00000000 | DEN\_PHY\_OBS\_REG\_0\_0: Controls loopback status, data and masking info for slice 0. |
| DENALI\_PHY\_08 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_0\_0: Reports DLL status for slice 0. |
| DENALI\_PHY\_09 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_1\_0: Reports DLL status for slice 0. |
| DENALI\_PHY\_10 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_2\_0: Reports DLL status for slice 0. |
| DENALI\_PHY\_11 | [31:0] | WR | 0x00000000 | PHY\_LVL\_DBG\_CONT\_REG\_0: Debug command to continue leveling sequence 0. |
| DENALI\_PHY\_12 | [31:0] | RW | 0x00000000 | PHY\_LVL\_CONFIG\_REG\_0: Controls lvl timings for slice 0. |
| DENALI\_PHY\_13 | [31:0] | RD | 0x00000000 | PHY\_GTLVL\_OBS\_REG\_0: Reports GTLVL status for slice 0. |
| DENALI\_PHY\_14 | [31:0] | RD | 0x00000000 | PHY\_RDLVL\_OBS\_REG\_0: Reports RDLVL status for slice 0. |
| DENALI\_PHY\_15 | [31:0] | RW | 0x00000000 | PHY\_WRLVL\_CONFIG\_REG\_0: Controls wrlvl timings for slice 0. |
| DENALI\_PHY\_16 | [31:0] | RD | 0x00000000 | PHY\_WRLVL\_OBS\_REG\_0: Reports WRLVL status for slice 0. |
| DENALI\_PHY\_17 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_18 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_19 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_20 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_21 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_22 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_23 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_24 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_25 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_26 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_27 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_28 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_29 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_30 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_31 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_32 | [31:0] |  | 0x00000000 | DEN\_PHY\_DQ\_TIMING\_REG\_1: Controls PHY slice 1. |
| DENALI\_PHY\_33 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQS\_TIMING\_REG\_1: Controls PHY slice 1. |
| DENALI\_PHY\_34 | [31:0] | RW | 0x00000000 | DEN\_PHY\_GATE\_LPBK\_CTRL\_REG\_1:  Controls PHY slice 1. |
| DENALI\_PHY\_35 | [31:0] | RW | 0x00000000 | DEN\_PHY\_READ\_CTRL\_REG\_1: Controls PHY slice 1. |
| DENALI\_PHY\_36 | [31:0] | RW | 0x00000000 | PHY\_DLL\_MASTER\_CTRL\_REG\_1: Controls the DLL for slice 1. |
| DENALI\_PHY\_37 | [31:0] | RW+ | 0x00000000 | PHY\_DLL\_SLAVE\_CTRL\_REG\_1: Controls the DLL for slice 1. |
| DENALI\_PHY\_38 | [31:0] | RW | 0x00000000 | DEN\_PHY\_IE\_TIMING\_REG\_1: Controls PHY slice 1. |
| DENALI\_PHY\_39 | [31:0] | RD | 0x00000000 | DEN\_PHY\_OBS\_REG\_0\_1: Controls loopback status, data and masking info for slice 1. |
| DENALI\_PHY\_40 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_0\_1: Reports DLL status for slice 1. |
| DENALI\_PHY\_41 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_1\_1: Reports DLL status for slice 1. |
| DENALI\_PHY\_42 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_2\_1: Reports DLL status for slice 1. |
| DENALI\_PHY\_43 | [31:0] | WR | 0x00000000 | PHY\_LVL\_DBG\_CONT\_REG\_1: Debug command to continue leveling sequence 1. |
| DENALI\_PHY\_44 | [31:0] | RW | 0x00000000 | PHY\_LVL\_CONFIG\_REG\_1: Controls lvl timings for slice 1. |
| DENALI\_PHY\_45 | [31:0] | RD | 0x00000000 | PHY\_GTLVL\_OBS\_REG\_1: Reports GTLVL status for slice 1. |
| DENALI\_PHY\_46 | [31:0] | RD | 0x00000000 | PHY\_RDLVL\_OBS\_REG\_1: Reports RDLVL status for slice 1. |
| DENALI\_PHY\_47 | [31:0] | RW | 0x00000000 | PHY\_WRLVL\_CONFIG\_REG\_1: Controls wrlvl timings for slice 1. |
| DENALI\_PHY\_48 | [31:0] | RD | 0x00000000 | PHY\_WRLVL\_OBS\_REG\_1:Reports WRLVL status for slice 1. |
| DENALI\_PHY\_49 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_50 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_51 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_52 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_53 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_54 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_55 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_56 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_57 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_58 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_59 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_60 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_61 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_62 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_63 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_64 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQ\_TIMING\_REG\_2: Controls PHY slice 2. |
| DENALI\_PHY\_65 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQS\_TIMING\_REG\_2: Controls PHY slice 2. |
| DENALI\_PHY\_66 | [31:0] | RW | 0x00000000 | DEN\_PHY\_GATE\_LPBK\_CTRL\_REG\_2: Controls PHY slice 2. |
| DENALI\_PHY\_67 | [31:0] | RW | 0x00000000 | DEN\_PHY\_READ\_CTRL\_REG\_2: Controls PHY slice 2. |
| DENALI\_PHY\_68 | [31:0] | RW | 0x00000000 | PHY\_DLL\_MASTER\_CTRL\_REG\_2: Controls the DLL for slice 2. |
| DENALI\_PHY\_69 | [31:0] | RW+ | 0x00000000 | PHY\_DLL\_SLAVE\_CTRL\_REG\_2: Controls the DLL for slice 2. |
| DENALI\_PHY\_70 | [31:0] | RW | 0x00000000 | DEN\_PHY\_IE\_TIMING\_REG\_2: Controls PHY slice 2. |
| DENALI\_PHY\_71 | [31:0] | RD | 0x00000000 | DEN\_PHY\_OBS\_REG\_0\_2: Controls loopback status, data and masking info for slice 2. |
| DENALI\_PHY\_72 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_0\_2: Reports DLL status for slice 2. |
| DENALI\_PHY\_73 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_1\_2: Reports DLL status for slice 2. |
| DENALI\_PHY\_74 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_2\_2: Reports DLL status for slice 2. |
| DENALI\_PHY\_75 | [31:0] | WR | 0x00000000 | PHY\_LVL\_DBG\_CONT\_REG\_2: Debug command to continue leveling sequence 2. |
| DENALI\_PHY\_76 | [31:0] | RW | 0x00000000 | PHY\_LVL\_CONFIG\_REG\_2: Controls lvl timings for slice 2. |
| DENALI\_PHY\_77 | [31:0] | RD | 0x00000000 | PHY\_GTLVL\_OBS\_REG\_2: Reports GTLVL status for slice 2. |
| DENALI\_PHY\_78 | [31:0] | RD | 0x00000000 | PHY\_RDLVL\_OBS\_REG\_2: Reports RDLVL status for slice 2. |
| DENALI\_PHY\_79 | [31:0] | RW | 0x00000000 | PHY\_WRLVL\_CONFIG\_REG\_2: Controls wrlvl timings for slice 2. |
| DENALI\_PHY\_80 | [31:0] | RD | 0x00000000 | PHY\_WRLVL\_OBS\_REG\_2: Reports WRLVL status for slice 2. |
| DENALI\_PHY\_81 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_82 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_83 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_84 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_85 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_86 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_87 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_88 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_89 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_90 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_91 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_92 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_93 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_94 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_95 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_96 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQ\_TIMING\_REG\_3: Controls PHY slice 3. |
| DENALI\_PHY\_97 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQS\_TIMING\_REG\_3：Controls PHY slice 3. |
| DENALI\_PHY\_98 | [31:0] | RW | 0x00000000 | DEN\_PHY\_GATE\_LPBK\_CTRL\_REG\_3：Controls PHY slice 3. |
| DENALI\_PHY\_99 | [31:0] | RW | 0x00000000 | DEN\_PHY\_READ\_CTRL\_REG\_3：Controls PHY slice 3. |
| DENALI\_PHY\_100 | [31:0] | RW | 0x00000000 | PHY\_DLL\_MASTER\_CTRL\_REG\_3：Controls the DLL for slice 3. |
| DENALI\_PHY\_101 | [31:0] | RW+ | 0x00000000 | PHY\_DLL\_SLAVE\_CTRL\_REG\_3：Controls the DLL for slice 3. |
| DENALI\_PHY\_102 | [31:0] | RW | 0x00000000 | DEN\_PHY\_IE\_TIMING\_REG\_3：Controls PHY slice 3. |
| DENALI\_PHY\_103 | [31:0] | RD | 0x00000000 | DEN\_PHY\_OBS\_REG\_0\_3：Controls loopback status, data and masking info for slice 3. |
| DENALI\_PHY\_104 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_0\_3：Reports DLL status for slice 3. |
| DENALI\_PHY\_105 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_1\_3：Reports DLL status for slice 3. |
| DENALI\_PHY\_106 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_2\_3：Reports DLL status for slice 3. |
| DENALI\_PHY\_107 | [31:0] | WR | 0x00000000 | PHY\_LVL\_DBG\_CONT\_REG\_3：Debug command to continue leveling sequence 3. |
| DENALI\_PHY\_108 | [31:0] | RW | 0x00000000 | PHY\_LVL\_CONFIG\_REG\_3：Controls lvl timings for slice 3. |
| DENALI\_PHY\_109 | [31:0] | RD | 0x00000000 | PHY\_GTLVL\_OBS\_REG\_3：Reports GTLVL status for slice 3. |
| DENALI\_PHY\_110 | [31:0] | RD | 0x00000000 | PHY\_RDLVL\_OBS\_REG\_3：Reports RDLVL status for slice 3. |
| DENALI\_PHY\_111 | [31:0] | RW | 0x00000000 | PHY\_WRLVL\_CONFIG\_REG\_3：Controls wrlvl timings for slice 3. |
| DENALI\_PHY\_112 | [31:0] | RD | 0x00000000 | PHY\_WRLVL\_OBS\_REG\_3：Reports WRLVL status for slice 3. |
| DENALI\_PHY\_113 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_114 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_115 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_116 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_117 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_118 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_119 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_120 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_121 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_122 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_123 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_124 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_125 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_126 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_127 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_128 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQ\_TIMING\_REG\_4：Controls PHY slice 4. |
| DENALI\_PHY\_129 | [31:0] | RW | 0x00000000 | DEN\_PHY\_DQS\_TIMING\_REG\_4：Controls PHY slice 4. |
| DENALI\_PHY\_130 | [31:0] | RW | 0x00000000 | DEN\_PHY\_GATE\_LPBK\_CTRL\_REG\_4：Controls PHY slice 4. |
| DENALI\_PHY\_131 | [31:0] | RW | 0x00000000 | DEN\_PHY\_READ\_CTRL\_REG\_4：Controls PHY slice 4. |
| DENALI\_PHY\_132 | [31:0] | RW | 0x00000000 | PHY\_DLL\_MASTER\_CTRL\_REG\_4：Controls the DLL for slice 4. |
| DENALI\_PHY\_133 | [31:0] | RW+ | 0x00000000 | PHY\_DLL\_SLAVE\_CTRL\_REG\_4：Controls the DLL for slice 4. |
| DENALI\_PHY\_134 | [31:0] | RW | 0x00000000 | DEN\_PHY\_IE\_TIMING\_REG\_4：Controls PHY slice 4. |
| DENALI\_PHY\_135 | [31:0] | RD | 0x00000000 | DEN\_PHY\_OBS\_REG\_0\_4：Controls loopback status, data and masking info for slice 4. |
| DENALI\_PHY\_136 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_0\_4：Reports DLL status for slice 4. |
| DENALI\_PHY\_137 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_1\_4：Reports DLL status for slice 4. |
| DENALI\_PHY\_138 | [31:0] | RD | 0x00000000 | PHY\_DLL\_OBS\_REG\_2\_4：Reports DLL status for slice 4. |
| DENALI\_PHY\_139 | [31:0] | WR | 0x00000000 | PHY\_LVL\_DBG\_CONT\_REG\_4：Debug command to continue leveling sequence 4. |
| DENALI\_PHY\_140 | [31:0] | RW | 0x00000000 | PHY\_LVL\_CONFIG\_REG\_4：Controls lvl timings for slice 4. |
| DENALI\_PHY\_141 | [31:0] | RD | 0x00000000 | PHY\_GTLVL\_OBS\_REG\_4：Reports GTLVL status for slice 4. |
| DENALI\_PHY\_142 | [31:0] | RD | 0x00000000 | PHY\_RDLVL\_OBS\_REG\_4：Reports RDLVL status for slice 4. |
| DENALI\_PHY\_143 | [31:0] | RW | 0x00000000 | PHY\_WRLVL\_CONFIG\_REG\_4：Controls wrlvl timings for slice 4. |
| DENALI\_PHY\_144 | [31:0] | RD | 0x00000000 | PHY\_WRLVL\_OBS\_REG\_4：Reports WRLVL status for slice 4. |
| DENALI\_PHY\_145 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_146 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_147 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_148 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_149 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_150 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_151 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_152 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_153 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_154 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_155 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_156 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_157 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_158 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_159 | [31:0] | - | - | This register intentionally blank. |
| DENALI\_PHY\_160 | [31:0] | RW | 0x00000000 | DEN\_PHY\_CTRL\_REG：Global controls for all slices |
| DENALI\_PHY\_161 | [31:1] | - | - | Reserve |
| 0 | WR | 0x0 | DEN\_PHY\_CALVL\_DEBUG\_STEP\_REG：CA leveling debug single step. |
| DENALI\_PHY\_162 | [31:0] | RW | 0x00000000 | DEN\_PHY\_CALVL\_CTRL\_REG：CA leveling controls for CA slice |
| DENALI\_PHY\_163 | [31:0] | RD | 0x00000000 | DEN\_PHY\_CALVL\_OBS\_REG：CA leveling observe values. |
| DENALI\_PHY\_164 | [31:0] | RW | 0x00000000 | DEN\_PHY\_LP\_WAKEUP\_REG：Specifies the number of cycles that PHY takes to wakeup in Low Power Mode. |
| DENALI\_PHY\_165 | [31:0] | RW | 0x00000000 | DEN\_PHY\_PAD\_TSEL\_REG：Global controls for all slices |
| DENALI\_PHY\_166 | [31:0] | RW | 0x00000000 | PHY\_PAD\_DRIVE\_REG\_0：Controls pad drive strengths |
| DENALI\_PHY\_167 | [31:0] | RW\_D | 0x00000000 | PHY\_PAD\_DRIVE\_REG\_1：Controls pad drive strengths |
| DENALI\_PHY\_168 | [31:0] | RW | 0x00000000 | PHY\_PAD\_DRIVE\_REG\_2：Controls pad drive strengths |
| DENALI\_PHY\_169 | [31:0] | RW\_D  + | 0x00070007 | PHY\_PAD\_TERM\_REG\_0：Controls pad termination values |
| DENALI\_PHY\_170 | [31:0] | RW\_D  + | 0x00070007 | PHY\_PAD\_TERM\_REG\_1：Controls pad termination values |
| DENALI\_PHY\_171 | [31:0] | RW | 0x00000000 | PHY\_PAD\_CAL\_CTRL\_REG\_0：Controls calibration block values |
| DENALI\_PHY\_172 | [31:0] | RD | 0x00000000 | PHY\_PAD\_CAL\_OBS\_REG\_0：Calibration block observation values. |
| DENALI\_PHY\_173 | [31:0] | RW | 0x00000000 | PHY\_PAD\_CAL\_INTERVAL\_COUNT\_0：Pad calibration interval counter compare value for block. |
| DENALI\_PHY\_174 | [31:8] | - | - | Reserve |
| [7:0] | RW | 0x00 | PHY\_PAD\_VREF\_CTRL\_REG\_AC：Pad VREF SLP\_RE\_000\_1215V port settings. |

* 1. Programming Sequence
     1. Controller Programming Sequence

The following figure is controller initial programming and MBIST programming:



Figure . Initial programming



Figure 42.MBIST programming

* + 1. PHY Programming Sequence

The following figure is PHY initial programming and loopback programming:



Figure .Initial program



Figure 44.Loopback programming

* 1. Programming Restrictions
     1. Controller Programming Restrictions
        1. Restrictions on the AXI Bus

For this Cadence DDR Controller, the full AXI specification is supported with the following restrictions:

• FIXED burst types are not supported.

• The response signals will never respond with a DECERR response type.

• Cacheable, read allocate or write allocate commands are not supported.

• Commands cannot cross a 4K boundary.

• WRAP commands must be aligned.

• WRAP commands must not cross a DRAM page boundary. To calculate the maximum AXI wrap size, use the following equation: N\*[AXI\_BUS\_WIDTH]/8 ≤Minimum DRAM Page Size, where N is a power of 2 from 1 to 16 (1, 2, 4, 8, 16) and the [AXI\_BUS\_WIDTH] is the port data width of the AXI port being analyzed.

• Write data cannot be interleaved. Write data must be presented to the Cadence DDR Controller in the same order as the write commands. Interleaved write data will result in undefined behavior.

• Each port may only monitor the exclusivity of a limited number of transactions at any time. In addition, each port will only maintain one exclusive monitor per source ID at a time.

• Locked access is not configured for this controller.

• The AHB register port only supports the AHB SINGLE burst type and transfer types of NONSEQ or IDLE

* + - 1. General LPDDR2 Restrictions

The Cadence DDR Controller does not support interleaving (MR1 [3]) and only supports memory burst lengths of 4 or 8. In addition, the MR1 [4] wrapping bit must be cleared to ’b0 meaning that data must be returned to the controller in memory wrap order. The controller will not take advantage of this functionality, but expects the data in this way. The burst terminatecommand and data not valid commands are not used.

1. I2C Controller






9. 1. Register Summary

Table 681.I2C Controller Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| I2C\_MTXR | 0x0000 | 0x00000000 | Master transmit register input |
| I2C\_MRXR | 0x0004 | 0x00000000 | Master receive register output |
| I2C\_STXR | 0x0008 | 0x00000000 | Slave transmit register input |
| I2C\_SRXR | 0x000C | 0x00000000 | Slave receive register output |
| I2C\_SADDR | 0x0010 | 0x000003FF | I2C controller slave address |
| I2C\_IER | 0x0014 | 0x00000000 | Enable/Mask interrupts generated by the I2C controller |
| I2C\_ISR | 0x0018 | 0x00000000 | Interrupt status register |
| I2C\_LCMR | 0x001C | 0x00000000 | I2C line command register |
| I2C\_LSR | 0x0020 | 0x00000000 | I2C core status |
| I2C\_CONR | 0x0024 | 0x00000000 | I2C operation register 1 |
| I2C\_OPR | 0x0028 | 0x00000000 | I2C operation register 2 |

* 1. Register Descriptions

I2C\_MTXR: This register contains data to be transmitted on the I2C for master purpose

Address: Operational Base + offset (0x00)

Table 682.I2C\_MTXR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | I2C master transmit data register. |

I2C\_MRXR :This register contains data to be received on the I2C for master purpose

Address: Operational Base + offset (0x04)

Table 683.I2C\_MRXR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | R | 0x0 | I2C master receive data register. |

I2C\_STXR: This register contains data to be transmitted on the I2C for slave purpose

Address: Operational Base + offset (0x08)

Table 684.I2C\_STXR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | I2C slave transmit data register. |

I2C\_SRXR: This register contains data to be received on the I2C for slave purpose

Address: Operational Base + offset (0x0C)

Table 685.I2C\_SRXR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | R | 0x0 | I2C slave receive data register. |

I2C\_SADDR: This register contains address to be matched on the I2C for slave purpose

Address: Operational Base + offset (0x10)

Table 686.I2C\_SADDR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bit | Type | | Reset Value | | Description | |
| 31:10 | | - | | - | | Reserved | |
| 9:0 | | RW | | 0x3FF | | Slave address | |

I2C\_IER: This register contains the bits to control the interrupt generation of I2C controller

Address: Operational Base + offset (0x14)

Table 687.I2C\_IER

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7 | RW | 0x0 | Arbitration lose interrupt enable bit.  “1” - enable. “0” - disable. |
| 6 | RW | 0x0 | Abnormal stop interrupt enable bit.  “1” - enable. “0” - disable. |
| 5 | RW | 0x0 | Broadcast address matches (address zero) interrupt enable bit.  “1” - enable. “0” - disable. |
| 4 | RW | 0x0 | Slave address matches interrupt enable bit.  “1” - enable. “0” – disable. |
| 3 | RW | 0x0 | Slave ACK period interrupt enable bit (SRX mode).  “1” - enable. “0” - disable. |
| 2 | RW | 0x0 | Slave receives ACK interrupt enable bit (STX mode).  “1” - enable. “0” - disable. |
| 1 | RW | 0x0 | Master ACK period interrupt enable bit (MRX mode).  “1” - enable. “0” - disable. |
| 0 | RW | 0x0 | Master receives ACK interrupt enable bit (MTX mode).  “1” - enable. “0” - disable. |

I2C\_ISR: I2C interrupt status register

Address: Operational Base + offset (0x18)

Table 688.I2C\_ISR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7 | RW | 0x0 | Arbitration lose status bit.  “1” – Arbitration lose occurs  “0” – No Arbitration lose occurs  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 6 | RW | 0x0 | Abnormal stop status bit.  “1” – Abnormal stop occurs  “0” – No abnormal stop occurs  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 5 | RW | 0x0 | Broadcast address (address zero) matches status bit.  “1” – Broadcast address matches.  “0” – No broadcast address matches.  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 4 | RW | 0x0 | Slave address matches status bit.  “1” –Slave address matches.  “0” – No slave address matches (When read).  Clear slave address matches interrupt (When write).  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 3 | RW | 0x0 | Slave ACK period interrupt status bit (SRX mode).  “1” – interrupt generation  “0” – no interrupt generation  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 2 | RW | 0x0 | Slave receives ACK interrupt status bit (STX mode).  “1” – interrupt generation  “0” – no interrupt generation  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 1 | RW | 0x0 | Master ACK period interrupt status bit (MRX mode).  “1” – interrupt generation  “0” – no interrupt generation  Write this bit “0” to clear.  Write “1” will not change this bit. |
| 0 | RW | 0x0 | Master receives ACK interrupt status bit (MTX mode).  “1” – interrupt generation  “0” – no interrupt generation  Write this bit “0” to clear.  Write “1” will not change this bit. |

I2C\_LCMR: This register contains the bits to generate start and stop commands of I2C controller

Address: Operational Base + offset (0x1C)

Table 689.I2C\_LCMR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:3 | - | - | Reserved |
| 2 | RW | 0x0 | “RESUME” condition generation bit.  “1” - enable. “0” - disable.  Write “1” to start “RESUME” action. It cannot be cancelled by write “0”. This bit is self-cleared after “RESUME” action.  Write “0” is undefined. |
| 1 | RW | 0x0 | “STOP” condition generation bit.  “1” - enable. “0” - disable.  Write “1” to start “STOP” action. It cannot be cancelled by write “0”. This bit is self-cleared after “STOP” action.  Write “0” is undefined. |
| 0 | RW | 0x0 | “START” condition generation bit.  “1” - enable. “0” - disable.  Write “1” to start “START” action. It cannot be cancelled by write “0”. This bit is self-cleared after “START” action.  Write “0” is undefined. |

I2C\_LSR: This register is used to read I2C core status

Address: Operational Base + offset (0x20)

Table 690.I2C\_LSR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:2 | - | - | Reserved |
| 1 | R | 0x0 | I2C receives ACK status bit (MTX and STX modes).  “0” – I2C bus receives ACK  “1” – I2C bus receives NAK. |
| 0 | R | 0x0 | I2C core busy status bit.  ‘1’ – After START condition detect.  ‘0’ – After STOP condition detect. |

I2C\_CONR: This register is used to set the operation modes and ACK enable bit of I2C controller.

Address: Operational Base + offset (0x24)

Table 691.I2C\_CONR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:5 | - | - | Reserved |
| 4 | RW | 0x0 | I2C bus acknowledge enable register  “0” – enable (ACK). “1” – disable (NAK).  When enable, the SDA is free (TX mode), and is LOW (RX mode) in acknowledge time. |
| 3 | RW | 0x0 | Master receive/transmit mode select bit  “0”: receive. “1”: transmit. |
| 2 | RW | 0x0 | Master port enable bit  “0”: disable. “1”: enable. |
| 1 | RW | 0x0 | Slave receive/transmit mode select bit  “0”: receive. “1”: transmit. |
| 0 | RW | 0x0 | Slave port enable bit  “0”: disable. “1”: enable |

I2C\_OPR: This register is used to set I2C core enable bit, frequency divider factor, internal state machine reset and slave address length modes

Address: Operational Base + offset (0x28)

Table 692.I2C\_OPR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:9 | - | - | Reserved |
| 8 | RW | 0x0 | I2C slave address mode bit.  “0” – 7 bits address mode.  “1” – 10 bits address mode. |
| 7 | RW | 0x0 | I2C state machine (both master/slave) reset bit.  “0” – don’t reset state machine  “1” – reset state machine |
| 6 | RW | 0x0 | I2C core enable bit  “0” – disable I2C controller.  “1” – enable I2C controller. |
| 5:0 | RW | 0x0 | I2C clock divisor bits (I2CCDVR).  The value of I2CCDVR is used to generate the transmit and receive bit rate of the I2C master part.  And the bit rate equation will be described more detail in section below. |

* 1. Programming Sequence

Operation

The I2C controller supports both the Master and Slave functions. It also supports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller is divided to 3 parts and described separately: initialization, master mode programming, and slave mode programming.

More details are listed in the Program Sequence section.

Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting & configuration must be conformed, which includes:

I2C Register memory mapping

I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.

I2C Clock Rate: The I2C controller uses the APB clock/5 as the system clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

Master Mode Programming

SCL Clock: When the I2C controller is programmed in Master mode, the SCL frequency is determine by I2C\_OPR register. The SCL frequency is calculated by the following formula

SCL Divisor = (I2CCDVR[5:3] + 1) x 2(I2CCDVR[2:0] + 1)

SCL = PCLK/ 5\*SCLK Divisor

The I2CCDVR[2:0] is coarse factor and I2CCDVR[5:3] is fine tune factor for the SCL clock.

Data Receiver Register Access: The Master data receive register (I2C\_MRXR) can only be correctly accessed at Master Receiver Mode. When accessing the I2C\_MRXR register, make sure the I2C\_CONR[3:2] is set to receiver mode.

Start Command and 1’st Byte Address: The I2C controller combines the start command and 1’st byte address data output together. SW cannot issue start command only. So before issuing start command, SW must prepare the correct address data (include R/W bit) to the I2C\_MTXR register. Since the I2C protocol allows the repeated start command, the resume command needs to be issued with repeated start.

Interrupt and Resume: I2C controller interrupt status is generated by HW and cleared by SW. The clear scheme is to write 0 to the correspond bit. Writing 1 to interrupt status bits will not get any affect. The interrupt clear affect only the interrupt status bits. For reasons of flexibility and interrupt latency reduction, the interrupt clear will not resume the I2C function. The I2C function resumes when I2C\_LCMR register resume bit is set to 1. This separates the I2C service routine from the ISR (Interrupt Service Routine). SW must carefully design the I2C service routine because the I2C controller may be locked in some special state. When operating at Transmit mode, SW should prepare the next transmit data on the I2C\_MTXR register before issue the resume command.

Read/Write Command: The I2C Read/Write command depends on the last bit of address. SW should take the responsibility of Read/Write control. The Read/Write control and the Master Receive/Transmit mode setting must be correctly set before resume the I2C function.

Multi-Master Arbitration: When I2C controller works on Multi-Master I2C bus, HW will detect the bus busy condition and arbitration loss. When it happens, HW will stop the transaction and notify SW. SW should take the responsibility of re-transmit and time-out handling.

Master Interrupt Condition: There are 3 interrupt bits in I2C\_ISR register related to master mode.

Master ACK (Bit 0): The bit is asserted when Master receives ACK. In other words, the interrupt happens only at Master Transmit Mode.

Master ACK Period (Bit 1): The bit is asserted when Master needs to send out ACK. In other words, the interrupt happens only at Master Receive Mode.

Arbitration Loss (Bit 7): The bit is asserted when Master starts a transaction but lose the bus arbitration.

Stop Command: Master can issue Stop command when receive Master ACK or Master ACK Period interrupt. Because the Stop command is attached at the end of a transaction, the resume command needs to be issued with stop command. According to the I2C spec, the NAK must be sent out at Receive Mode in Master ACK Period before Stop.

Slave Mode Programming

Data Receiver Register Access: The Slave data receive register (I2C\_SRXR) can only be correctly accessed at Slave Receiver Mode. When accessing the I2C\_SRXR register, make sure the I2C\_CONR[1:0] is set to slave mode.

7 Bits and 10 Bits Address: I2C Slave transaction starts when Slave address is matched. The I2C controller supports both the standard 7 bits address mode and 10 bits address mode. The I2C controller filters out the transaction of which address is not matched with the I2C\_SADDR register. However, I2C controller only filters the 1st address mode, SW should take care the 2nd address for 10 bits address mode. The 7 or 10 bits address mode is set with I2C\_OPR[8].

7 Bits Address Setting: Slave function begins upon detecting a received address matched with I2C\_SADDR register. The I2C\_SADDR does not include the Read/Write bit. The I2C\_SADDR[9:7] is ignored at 7 bits address mode.

10 Bits Address Setting: The I2C\_SADDR register must be correctly initialized before slave function start to work. The I2C\_SADDR does not include the Read/Write bit. The I2C controller detects the received 1st address by the I2C\_SADDR[9:8] combined with the 10 bits mode address prefix(0b11110xx).

Address Matching: The 1st transaction received by I2C controller in slave mode is the address. When the address matched with the slave address of the I2C controller, it will notify SW with an interrupt. I2C\_ISR[4] high represents the incoming slave address matched with the specific slave address of the I2C controller. I2C\_ISR[5] high represents the broadcast, the general call (0x00), is detected.

When address matched, SW should read the I2C\_SRXR register to figure out the transaction is a read or write and set the slave mode accordingly before resume ACK. If the next transaction is a read, the read data needs to be prepared to the I2C\_STXR before resume.

10 Bits Address 2nd Phase: Because the I2C controller takes care only the 1st address matching at 10 bits address mode, SW should take care the 2nd address comparison. When the 2nd byte address comparison fails, SW should issue a reset, I2C\_OPR[7], and issue a resume. After reset and resume, HW will ignore the rest coming transactions until next start detected.

Interrupt and Resume: the interrupt is generator by I2C controller and cleared by SW. The clear scheme is to write 0 to the corresponding bit. Writing 1 to interrupt status bits will not get any affect. The interrupt clear affect only the interrupt status bits. For reasons of flexibility and interrupt latency reduction, the interrupt clear will not resume the I2C function. The I2C function resumes when I2C\_LCMR register resume bit is set to 1. This separates the I2C service routine from the ISR (Interrupt Service Routine). SW must carefully design the I2C service routine because the I2C controller may be locked in some special state. When operating at Transmit mode, SW should prepare the next transmit data on the I2C\_STXR register before issue the resume command.

Read/Write Command: The I2C Read/Write command depends on the last bit of address. SW should take the responsibility of Read/Write control. The Read/Write control and the Master Receive/Transmit mode setting must be correctly set before resume the I2C function.

Slave Interrupt Condition: There are 5 interrupt bits in I2C\_ISR register related to slave mode.

Slave ACK (Bit 2): The bit is asserted when Slave receives ACK. In other words, this interrupt happens only at Salve Transmit Mode.

Salve ACK Period (Bit 3): The bit is asserted when Slave needs to send out ACK. In other words, this interrupt happens only at Slave Receive Mode.

Slave address match (bit 4): The bit is asserted when the coming address is matched with the slave address of the I2C controller. Slave ACK interrupt is not asserted when Slave address match interrupt is asserted.

Broadcast address match (bit 5): The bit is asserted when the coming address matched with general call (0x00) address. Slave ACK interrupt is not asserted when general call address match interrupt is asserted.

Abnormal stop occurs (bit 6): The bit is asserted when Slave receive abnormal stop.

I2C controller data transfer waveform

Bit transferring

(a) Data Validity

The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.



(b) START and STOP conditions

START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.



(a) Acknowledge

After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means “ACK”, on the contrary, it’s “NOT ACK”.



(b) Byte transfer

The master own I2C bus might initiate multi byte to transfers to a slave, the transfers starts from a “START” command and ends in a “STOP” command. After every byte transfer, the receiver must reply an ACK to transmitter.



(c)Arbitration

Arbitration takes place at SDA line, while the SCL line is at high level. The master transmits a high level, while another master transmits a low level will lose arbitration.



(d) Synchronization

Clock synchronization is performed using the wired-and connextion of I2C interface to the SCL line.





Figure 45.Master/Transmitter Mode Operations



Figure 46.Master/Receiver Mode Operations



Figure 47.Slave/Transmitter Mode Operations



Figure 48.Slave/Transmitter Mode Operations

* 1. Programming Restrictions

No programming restriction

1. SPI Controller
   1. Register Summary

Table 693.SPI Controller Register summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Offset | Size | Reset Value | Description |
| SPI\_TxR | 0x0000 | W | 0x00000000 | SPI master/slave controller transmit FIFO input |
| SPI\_RxR | 0x0000 | W | 0x00000000 | SPI master/slave controller receiver FIFO output |
| SPI\_IER | 0x0004 | W | 0x00000000 | Enable/Mask interrupts generated by the SPI master controller |
| SPI\_FCR | 0x0008 | W | 0x00000000 | SPI master/slave controller FIFO control register |
| SPI\_FWCR | 0x000C | W | 0x00000100 | SPI master/slave controller transaction flow control register |
| SPI\_DLYCR | 0x0010 | W | 0x00000000 | SPI master controller delay control register (master only) |
| SPI\_TxCR | 0x0014 | W | 0x00000000 | Transmit counter (master only) |
| SPI\_RxCR | 0x0018 | W | 0x00000000 | Receive counter (master only) |
| SPI\_SSCR | 0x001C | W | 0x00000000 | SPI master/slave controller slave select and characteristic register |
| SPI\_ISR | 0x0020 | W | 0x00000000 | SPI master/slave controller interrupt status register |
| SPI\_FIFO\_STAT | 0x0024 | W | 0x00000011 | SPI Transmit FIFO and Receive FIFO status |
| SPI\_TX\_REG0 | 0x0028 | W | 0x00000000 | SPI transmit register 0 for normal usage |
| SPI\_TX\_REG1 | 0x002C | W | 0x00000000 | SPI transmit register 1 for normal usage |
| SPI\_TX\_REG2 | 0x0030 | W | 0x00000000 | SPI transmit register 2 for normal usage |
| SPI\_TX\_REG3 | 0x0034 | W | 0x00000000 | SPI transmit register 3 for normal usage |
| SPI\_RX\_REG0 | 0x0038 | W | 0x00000000 | SPI receive register 0 for normal usage |
| SPI\_RX\_REG1 | 0x003C | W | 0x00000000 | SPI receive register 1 for normal usage |
| SPI\_RX\_REG2 | 0x0040 | W | 0x00000000 | SPI receive register 2 for normal usage |
| SPI\_RX\_REG3 | 0x0044 | W | 0x00000000 | SPI receive register 3 for normal usage |

* 1. Register Description

SPI\_TxR: This register contains data to be transmitted on the SPI master/slave controller bus on the MOSI pin

Address: Operational Base + offset (0x0000)

Table 694.SPI\_TxR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved |
| 15:0 | WO | 0x0 | SPI master/slave controller transmit data FIFO input. |

SPI\_RxR:This register contains data received from the SPI master/slave controller bus on the MISO pin

Address: Operational Base + offset (0x00)

Table 695.SPI\_RxR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved |
| 15:0 | RO | 0x0 | SPI master/slave controller receive data FIFO output. |

SPI\_IER: This register contains bits for controlling the interrupt generation of the SPI master/slave controller

Address: Operational Base + offset (0x04)

Table 696.SPI\_IER

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:12 | - | - | Reserved |
| 11 | RO | 0x0 | Character length mismatch interrupt enable bit. (CHARLNIEN) (slave only)  The bit enables character length mismatch interrupt. When finishing transfer, the received data that character length mismatch the setting of SPI\_SSCR[14:11].  “1” Enabled.  “0” Disabled. |
| 10 | RW | 0x0 | Rx register 3 data available interrupt enable bit. (RXREG3IEN) (slave only)  The bit enables the Rx register 3 data available interrupt when one piece of data is available in Rx register 3.  “1” Enabled.  “0” Disabled. |
| 9 | RW | 0x0 | Rx register 2 data available interrupt enable bit. (RXREG2IEN) (slave only)  The bit enables the Rx register 2 data available interrupt when one piece of data is available in Rx register 2.  “1” Enabled.  “0” Disabled. |
| 8 | RW | 0x0 | Rx register 1 data available interrupt enable bit. (RXREG1IEN) (slave only)  The bit enables the Rx register 1 data available interrupt when one piece of data is available in Rx register 1.  “1” Enabled.  “0” Disabled. |
| 7 | RW | 0x0 | Rx register 0 data available interrupt enable bit. (RXREG0IEN) (Slave only)  The bit enables the Rx register 0 data available interrupt when one piece of data is available in Rx register 0.  “1” Enabled.  “0” Disabled. |
| 6 | RW | 0x0 | Transmit FIFO empty interrupt enable bit. (TXFEIEN)  The bit enables transmit FIFO interrupt when transmit FIFO is empty.  “1” Enabled.  “0” Disabled. |
| 5 | RW | 0x0 | Slave select signal rising detection enable bit. (SSNRIEN) (Slave only)  The bit enables the slave select signal rising detection interrupt when the master set slave select bit changes from “0” to “1”.  “1” Enabled.  “0” Disabled. |
| 4 | RW | 0x0 | Receive data available enable bit. (RXAVIEN)  The bit enables receive FIFO interrupt when at least one data available in receive FIFO.  “1” Enabled.  “0” Disabled. |
| 3 | RW | 0x0 | Transmit FIFO interrupt enable bit (TxFIEN). This bit enables the transmit FIFO interrupt when the transmit FIFO trigger level is reached.  “1” Enabled.  “0” Disabled. |
| 2 | RW | 0x0 | Receive FIFO interrupt enable bit (RxFIEN). This bit enables the receive FIFO interrupt when the receive FIFO trigger level is reached.  “1” Enabled.  “0” Disabled. |
| 1 | RW | 0x0 | Receive FIFO overrun interrupt enable bit (RxFOIEN). This bit enables the receive FIFO overrun interrupt when the receive FIFO overrun condition has occurred.  “1” Enabled.  “0” Disabled. |
| 0 | RW | 0x0 | Receive transfer complete interrupt enable bit (RxCIEN). (master only)  This bit enables the receive transfer complete interrupt each time a receive transaction has ended.  “1” Enabled.  “0” Disabled. |

SPI\_FCR: The FCR allows selection of the FIFO trigger level (the number of entries in the receive FIFO required to enable the receive FIFO interrupt and the number of empty entries in the transmit FIFO required to enable the transmit FIFO interrupt). The FIFOs can also be cleared using this register

Address: Operational Base + offset (0x08)

Table 697.SPI\_FCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:14 | - | - | Reserved |
| 13:11 | RW | 0x0 | Defines the receive FIFO interrupt trigger level.  The receive FIFO interrupt trigger level meaning is described below.  For example, trigger level 4 entries indicate that there are at 4 data available entries in the receive FIFO.  0x0 – 2 entries 0x1 – 4 entries  0x2 – 6 entries  0x3~0x7 – Reserved. |
| 10:8 | RW | 0x0 | Define the transmit FIFO interrupt trigger level.  The transmit FIFO interrupt trigger level meaning is described below.  For example, trigger level 4 entries indicate that there are at least 4 empty location entries for pushing data into the transmit FIFO.  0x0 – 2 entries 0x1 – 4 entries  0x2 – 6 entries  0x3~0x7 – Reserved. |
| 7:4 | - | - | Reserved. |
| 3 | WO | 0x0 | SPI master/slave controller receive FIFO reset bit (CLRRXF\_N).  Writing a ‘1’ to this bit resets the receive FIFO. |
| 2 | WO | 0x0 | SPI master/slave controller transmit FIFO reset bit (CLRTXF\_N).  Writing a ‘1’ to this bit resets the transmit FIFO. |
| 1 | RO | 0x0 | Transmit FIFO full flag (TxFF).  This bit is set whenever the transmit FIFO is full. |
| 0 | RO | 0x0 | Receive data available flag (RxDAF).  This bit is set whenever at least has one data entry is available in the receive FIFO. |

Notes:The transmit and receive FIFO reset signals sustain 3 pclk cycles. During the reset period, any access to FIFO is ignored.

SPI\_FWCR: This register is used to control SPI master/slave controller transaction flow

Address: Operational Base + offset (0x0C)

Table 698.SPI\_FWCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:15 | - | - | Reserved |
| 14 | RW | 0x0 | Slave select signal control enable bit (SSC) (master only)  0: Slave select signal is controlled by HW  1: Slave select signal is controlled by SW |
| 13 | RW | 0x1 | Slave select signal active bit (SSA) (master R/W, and slave read only)  Writing a ‘1’ to this bit will set the slave select signal not active. The value of slave select signal changes from ‘0’ to ‘1’.  Writing a ‘0’ to this bit will set the slave select signal active. The value of the slave select signal changes from ‘1’ to ‘0’. |
| 12 | RW | 0x0 | SPI master/slave mode (SPIMD). This bit selects the SPI master/slave mode.  “0” SPI is in Slave mode  “1” SPI is in Master mode |
| 11 | WO | 0x0 | SPI master/slave controller soft reset bit (SRST\_N). Writing a ‘1’ to this bit resets the SPI master/slave controller logic. |
| 10 | RW | 0x0 | SPI master/slave enable bit (SPIEN). This bit enables the SPI master controller.  “1” SPI master/slave enabled  “0” SPI master/slave disabled |
| 9 | RW | 0x0 | SPI run bit (SPIRUN). (Master only)  When the CPU sets this bit from “0” to “1”, the SPI master/slave controller begins to transfer the data stored in the transmit FIFO and/or receive the data into the receive FIFO on the SPI master/slave controller bus. This will automatically be cleared to “0” after the transaction has ended. |
| 8 | RW | 0x1 | SPI master controller clock idle enable bit (CKIDLEN). (Master only)  This bit determines whether or not the SPI master controller clock can be asserted in an idle state during a transaction process, assuming that the master core meets the receive FIFO full or transmit FIFO empty conditions.  “1” SPI master controller clock can be asserted in an idle state.  “0” SPI master controller clock cannot be asserted in an idle state. |
| 7 | - | - | Reserved. |
| 6 | RW | 0x0 | Hardware DMA request enable |
| 5 | RW | 0x0 | Transmit and receive simultaneously transfer enable (TxRxsten) (master only)  This bit indicates whether or not the transmit and receive transfer can occur concurrently during a transaction.  “0” Tx and Rx cannot concurrently happen.  ”1” Tx and Rx can concurrently happen. |
| 4 | RW | 0x0 | SPI master/slave controller clock polarity bit (CPOL). This bit determines the polarity of the SCLK.  “0” SCLK is low when idle.  “1” SCLK is high when idle. |
| 3 | RW | 0x0 | Clock Phase Bit (CPHA). This bit determines the clock phase of the SCLK in relationship to the serial data.  "0" data is valid on the first SCLK edge (rising or falling) after slave select has asserted.  "1" data is valid on the second SCLK edge (rising or falling) after slave select has asserted. |
| 2 | RW | 0x0 | LSB-First Enable (LSBEN).  “1” LSB first transfer.  “0” MSB first transfer. |
| 1 | - | - | Reserved. |
| 0 | RW | 0x0 | Loop back mode (LBKMD) (master only)  This bit is indicates the operation mode of the SPI master controller is in a normal operation mode or in a loop back mode.  ‘0’ Normal operation mode.  ‘1’ Loop back mode. |

Notes: The soft reset signal can sustain 3 pclk cycles. During the reset period, any access to FIFO is ignored.

SPI\_DLYCR :This register sets the necessary clock delay during a transaction according to specific slave device requirements

Address: Operational Base + offset (0x10)

Table 699.SPI\_DLYCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:11 | - | - | Reserved |
| 10:8 | RW | 0x0 | Period between Tx and Rx transfer (PBTxRx). This field defines the delay between transmit transfer complete and receive transfer start.  0x0: Non-SPI master controller clock delay.  0x1: 4 SPI master controller clock delay.  0x2: 8 SPI master controller clock delay.  0x3: 16 SPI master controller clock delay.  0x4: 32 SPI master controller clock delay.  0x5: 64SPI master controller clock delay.  0x6: 128 SPI master controller clock delay.  0x7: 256 SPI master controller clock delay. |
| 7:6 | - | - | Reserved. |
| 5:3 | RW | 0x0 | Period between two consecutive transfers (PBCT). This field defines the delay between consecutive transfers to the device without removing its chip select.  0x0: Non-SPI master controller clock delay.  0x1: 4 SPI master controller clock delay.  0x2: 8 SPI master controller clock delay.  0x3: 16 SPI master controller clock delay.  0x4: 32 SPI master controller clock delay.  0x5: 64 SPI master controller clock delay.  0x6: 128 SPI master controller clock delay.  0x7: 256 SPI master controller clock delay. |
| 2:0 | RW | 0x0 | Period before SPI master controller clock active (PBCA). This field defines the delay before the SPI master controller clock changes from idle to active after the chip select is asserted.  0x0: 1/2 SPI master controller clock delay.  0x1: 4 SPI master controller clock delay.  0x2: 8 SPI master controller clock delay.  0x3: 16 SPI master controller clock delay.  0x4: 32 SPI master controller clock delay.  0x5: 64SPI master controller clock delay.  0x6: 128 SPI master controller clock delay.  0x7: 256 SPI master controller clock delay. |

Notes: The timing diagram below illustrates the delay meaning.



Figure 49.SPI clock’s delay meaning

SPI\_TxCR: This register controls the total transfer data size in each transmit transaction

Address: Operational Base + offset (0x0014)

Table 700.SPI\_TxCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved |
| 15:0 | RW | 0x0 | 0: Stop the transmit transaction.  1-65535: Start a transmit transaction. |

SPI\_RxCR: This register controls the total transfer data size in each receive transaction

Address: Operational Base + offset (0x0018)

Table 701.SPI\_RxCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved. |
| 15:0 | RW | 0x0 | 0: Stop the receive transaction.  1-65535: Start a receive transaction. |

SPI\_SSCR: SPI master/slave controller slaves select and characteristic control register

Address: Operational Base + offset (0x001C)

Table 702.SPI\_SSCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:15 | - | - | Reserved. |
| 14:11 | RW | 0x0 | Character length determines bits (SPICHRL).  This field specifies how many bits are to be transferred in each transfer.  0x0: Reserved  0x1: Reserved  0x2: Reserved  0x3: 4 bits (master only)  0x4: 5 bits (master only)  0x5: 6 bits (master only)  0x6: 7 bits (master only)  0x7: 8 bits  0x8: 9 bits  0x9: 10 bits  0xA: 11 bits  0xB: 12 bits  0xC: 13 bits  0xD: 14 bits  0xE: 15 bits  0xF: 16 bits |
| 10:8 | RW | 0x0 | SPI master controller slave select register (SPISSR) (master only)  0x0: Slave0 device (SPIx\_SSN[0]) is set to the active state.  0x1: Slave1 device (SPIx\_SSN[1]) is set to the active state.  0x2: Slave2 device (SPIx\_SSN[2]) is set to the active state.  0x3: Slave3 device (SPIx\_SSN[3]) is set to the active state.  0x4: Slave4 device (SPIx\_SSN[4]) is set to the active state.  0x5: Slave5 device (SPIx\_SSN[5]) is set to the active state.  0x6: Slave6 device (SPIx\_SSN[6]) is set to the active state.  0x7: Slave7 device (SPIx\_SSN[7]) is set to the active state. |
| 7:6 | - | - | Reserved. |
| 5:0 | RW | 0x0 | SPI master controller clock divisor bits (SPIDIVR) (master only)  The value of SPIDIVR is used to generate the transmit and receive bit rate of the SPI master controller. The following section describes the bit rate equation in more detail. |

Notes: CPHA, CPOL, should be set to match the protocol expected by the SPI slave device.

SPI\_ISR: SPI master/slave controller interrupt status register

Address: Operational Base + offset (0x0020)

Table 703.SPI\_ISR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:12 | - | - | Reserved. |
| 11 | RW | 0x0 | Character length mismatch interrupt flag (CHARLNIF) (slave only).  While finishing transfer, the received data that character length mismatch the setting of SPI\_SSCR[14:11], the bit is set.  An interrupt will be asserted to the CPU when this bit is set and the CHARLNIEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 10 | RW | 0x0 | Rx register 3 data available interrupt flag (RXREG3IF) (slave only).  This bit is set when data available in Rx register 3. An interrupt will be asserted to the CPU when this bit is set and the RXREG3IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 9 | RW | 0x0 | Rx register 2 data available interrupt flag (RXREG2IF) (slave only).  This bit is set when data is available in Rx register 2. An interrupt will be asserted to the CPU when this bit is set and the RXREG2IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 8 | RW | 0x0 | Rx register 1 data available interrupt flag (RXREG1IF) (slave only).  This bit is set when data is available in Rx register 1. An interrupt will be asserted to the CPU when this bit is set and the RXREG1IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 7 | RW | 0x0 | Rx register 0 data available interrupt flag (RXREG0IF) (slave only).  This bit is set when data is available in Rx register 0. An interrupt will be asserted to the CPU when this bit is set and the RXREG0IEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 6 | RO | 0x0 | Transmit FIFO empty interrupt flag (TXFEIF).  This bit is set when the transmit FIFO is empty. An interrupt will be asserted to the CPU when this bit is set and the TXFEIEN bit is enabled. This bit will be cleared when the transmit FIFO is not empty. |
| 5 | RW | 0x0 | Slave select signal rising interrupt flag (SSNRIF) (slave only).  This bit is set when the slave select signal is asserted. An interrupt will be asserted to the CPU when this bit is set and the SSNRIEN bit is enabled. The bit is cleared by writing “1” to the register. |
| 4 | RO | 0x0 | Receive data available interrupt flag (RXAVIF).  This bit is set when at least one piece of data is available in receive FIFO. An interrupt will be asserted to the CPU when this bit is set and the RXAVIEN bit is enabled. This bit will be cleared when the receive FIFO is empty. |
| 3 | RO | 0x0 | Transmit FIFO interrupt flag (TxFIF).  This bit is set when the transmit FIFO trigger level is reached, and CPU wishes to keep transmit data to the device. An interrupt will be asserted to the CPU when this bit is set and the TxFIEN bit is enabled. This bit will be cleared when the transmit FIFO pointer over the trigger level. |
| 2 | RO | 0x0 | Receive FIFO interrupt flag (RxFIF).  This bit is set whenever the receive FIFO trigger level is reached. An interrupt will be asserted to the CPU when this bit is set and the RxFIEN bit is enabled. This bit will be cleared when the receive FIFO pointer drops below the trigger level. |
| 1 | RO | 0x0 | SPI master/slave controller overrun interrupt flag (SPIORIF).  ‘1’ – If the receive FIFO is full and another character has been received in the receiver shift register. If another character is starting to arrive, it will overwrite the data in the shift register, but the FIFO will remain intact. The bit is cleared if the receive FIFO is cleared by software simultaneously.  ‘0’ – No overrun state an another SPI master controller transaction |
| 0 | RO | 0x0 | Receive complete interrupt flag (RxCIF) (master only).  This bit is set whenever the receive transaction is over. An interrupt will be asserted to the CPU when this bit is set and the RxCIEN bit is enabled. The bit is cleared upon reading from the register. |

SPI\_FIFO\_STAT: This register contains the SPI Transmit FIFO and Receive FIFO status

Address: Operational Base + offset (0x0024)

Table 704.SPI\_ FIFO\_STAT

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:7 | - | - | Reserved. |
| 6 | RO | 0x0 | Receive FIFO full bit. (RxFF\_FULL)  This bit shows whether or not the Receive FIFO is full.  “1”: Receive FIFO is full.  “0”: Receive FIFO is not full. |
| 5 | RO | 0x0 | Receive FIFO half full bit. (RxFF\_HALFFULL)  This bit shows whether or not the Receive FIFO is half full.  “1”: There are four pieces of data in the Receive FIFO.  “0”: There are not four pieces of data in the Receive FIFO. |
| 4 | RO | 0x1 | Receive FIFO empty bit. (RxFF\_EMPTY)  This bit shows whether or not the Receive FIFO is empty.  “1”: Receive FIFO is empty.  “0”: Receive FIFO is not empty. |
| 3 | - | - | Reserved. |
| 2 | RO | 0x0 | Transmit FIFO full bit. (TxFF\_FULL)  This bit shows whether or not the transmit FIFO is full.  “1”: Transmit FIFO is full.  “0”: Transmit FIFO is not full. |
| 1 | RO | 0x0 | Transmit FIFO half full bit. (TxFF\_HALFFULL)  This bit shows whether or not the transmit FIFO is half full.  “1”: There are four or more pieces of data in the Transmit FIFO.  “0”: There are not four pieces of data in the Transmit FIFO. |
| 0 | RO | 0x1 | Transmit FIFO empty bit. (TxFF\_EMPTY)  This bit shows whether or not the transmit FIFO is empty.  “1”: Transmit FIFO is empty.  “0”: Transmit FIFO is not empty. |

Note: The half full bit will also be “1” when FIFO if full.

SPI\_TX\_REGx (x = 0, 1, 2, 3): This register contains data to be transmitted on the SPI controller bus on the MISO pin

Address: Operational Base + offset (0x0028/0x002C/0x0030/0x0034)

Table 705.SPI\_ TX\_REGx (x = 0, 1, 2, 3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved. |
| 15:0 | RW | 0x0 | SPI master/slave controller transmit data register (slave only) |

SPI\_RX\_REGx (x = 0, 1, 2, 3): This register contains the data received from the SPI controller bus on the MOSI pin

Address: Operational Base + offset (0x0038/0x003C/0x0040/0x0044)

Table 706.SPI\_ RX\_REGx (x = 0, 1, 2, 3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:16 | - | - | Reserved. |
| 15:0 | RO | 0x0 | SPI master/slave controller receive data register (slave only) |

* 1. Programming Sequence

The SPI Slave receives HW commands from the Master before transferring data.

The following table contains a list of these instructions and their operation codes. All instructions start at a high-to-low pcs\_n transition.

Table 707.Hardware command

|  |  |
| --- | --- |
| Command | Description |
| 0x01 | MISO data from Slave Tx FIFO |
| 0x02 | MISO data from Slave Tx/Rx FIFO status |
| 0x03 | MISO data from Slave Tx register 0 |
| 0x04 | MISO data from Slave Tx register 1 |
| 0x05 | MISO data from Slave Tx register 2 |
| 0x06 | MISO data from Slave Tx register 3 |
| 0x08 | MOSI data to Slave Rx FIFO |
| 0x09 | MOSI data to Slave Rx register 0 |
| 0x0A | MOSI data to Slave Rx register 1 |
| 0x0B | MOSI data to Slave Rx register 2 |
| 0x0C | MOSI data to Slave Rx register 3 |

Note: The command length depends on SPI\_SSCR[14:11]. The minimum size is 8 bits. For example, if the bit length is 16 bits, the master should transmit 0x0008 to the slave to write data to the slave Rx FIFO.

* + 1. HW command timing

Command 0x01

圖片2

Figure 50.Command 0x01

Command 0x02

圖片2

Figure 51.Command 0x02

Command 0x03 ~ Command 0x06



SLCK

MOSI

MISO

SS\_N

0x03-0x06

Data

Figure 52.Command 0x03 ~ Command 0x06

Command 0x08

圖片2

Figure 53.Command 0x08

Command 0x09 ~ Command 0x0C



SLCK

MOSI

MISO

SS\_N

0x09-0x0c

Data

Figure 54.Command 0x09 ~ Command 0x0C

* + 1. SPI master controller operation flow chart

The flow chart below describes how the software configure and performs a SPI master controller transaction through the SPI master/slave controller.



Figure 55.SPI master controller transaction

* + 1. SPI slave controller operation flow chart

The flow chart below describes how the software configures and performs a SPI slave controller transaction through the SPI master/slave controller.

圖片2

Figure 56.SPI slave controller transaction

* 1. Programming Restrictions

No programming restriction

1. UART Controller
   1. Registers Summary

Table 708.UART Controller Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| UART\_RBR | 0x0000 | 0x00000000 | Receiver FIFO output |
| UART\_THR | 0x0000 | - | Transmit poll demand |
| UART\_IER | 0x0004 | 0x00000000 | Enables/Masks interrupts generated by the UART. |
| UART\_IIR | 0x0008 | 0x000000C1 | Gets interrupt inform ation |
| UART\_FCR | 0x0008 | 0x000000C0 | Controls FIFO options |
| UART\_LCR | 0x000C | 0x00000003 | Line Control register. |
| UART\_MCR | 0x0010 | 0x00000000 | Modem Controls register |
| UART\_LSR | 0x0014 | 0x00000060 | Line Status information. |
| UART\_MSR | 0x0018 | 0x00000000 | Modem Status. |
| UART\_ICR | 0x001C | 0x00000000 | IrDA control register. |
| UART\_CTRLR | 0x0020 | 0x00000000 | UART baud rate clock source selection. |
| UART\_DMA\_MODE | 0x0024 | 0x00000000 | DMA mode select |

Note: In addition, there are 2 Clock Divisor registers that together form one 16-bit register.

These registers can be accessed when the 7th (DLAB) bit of the Line Control Register is set to ‘1’. At this

time, the above registers at addresses 0x00 and 0x04 cannot be accessed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Offset | Size | Reset Value | Description |
| UART\_DLL | 0x0000 | W | 0x00000000 | The LSB of the divisor latch. |
| UART\_DLH | 0x0004 | W | 0x00000000 | The MSB of the divisor latch |

* 1. Registers Descriptions

UART\_RBR

Address: Operational Base + offset (0x0000)

Receive Buffer Register

Table 709.UART\_RBR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | R | 0x0 | Any data words received by the UART form the serial link are accessed by reading this register.  Bit 0 in the LSR line status register can be used to check if all received bytes have been read. This bit will change to zero if no more bytes are present. |

UART\_THR:Transmitter holding register

Address: Operational Base + offset (0x0000)

Table 710.UART\_THR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | W | - | This register is used to buffer outgoing characters. Bit 5 in the line status register can be used to check if new information must be written to THR. A value of 1 indicates that the register is empty. More than one character can be written to the transmitter holding register when the bit signals an empty state. |

UART\_IER: Interrupt enable register enables disables UART interrupt generation

Address: Operational Base + offset (0x0004)

Table 711.UART\_IER

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:4 | - | - | Reserved. |
| 3 | RW | 0x0 | Modem status Interrupt  ‘0’ – Disabled ‘1’ – Enabled |
| 2 | RW | 0x0 | Receiver line status interrupt  ‘0’ – Disabled ‘1’ – Enabled |
| 1 | RW | 0x0 | Transmitter holding register empty interrupt  ‘0’ – Disabled ‘1’ – Enabled |
| 0 | RW | 0x0 | Received data available interrupt  ‘0’ – Disabled ‘1’ – Enabled |

UART\_IIR: This interrupt identification register enables the programmer to retrieve the current highest priority pending interrupt

Address: Operational Base + offset (0x0008)

Table 712.UART\_IIR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7:6 | R | 0x3 | FIFO status  00: No FIFIO  10: Unusable FIFO  01: NA  11: FIFO enable  Present for compatibility of 16550.  Since our UART will enable FIFO permanently, the FIFO status will always be “11” |
| 5 | R | 0x0 | Indicates whether TX circuit be sw-reset or not.  ‘0’ – Non-executing software reset.  ‘1’ – Executing software reset now. |
| 4 | R | 0x0 | Indicates whether RX circuit be sw-reset or not.  ‘0’ – Non-executing software reset.  ‘1’ – Executing software reset now. |
| 3:1 | R | 0x0 | Indicates the current highest priority pending interrupt. |
| 0 | R | 0x1 | Indicates that an interrupt is pending or not.  ‘0’ – An interrupt is pending.  ‘1’ – No interrupt is pending. |

The following table displays the list of possible interrupts along with the bits they enable, their priority levels, and their source and reset controls.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bit 3 | Bit 2 | Bit 1 | Priority | Interrupt Type | Interrupt Source | Interrupt Clear Control |
| 0 | 1 | 1 | 1st | Receiver line status | Parity, Overrun or Framing errors or Break Interrupt. | Reading the Line Status Register. |
| 0 | 1 | 0 | 2nd | Receive Data available | FIFO trigger level reached. | FIFO trigger level un-reached. |
| 1 | 1 | 0 | 3rd | Timeout indication | There is at least 1 character in the FIFO but no character has been input to the FIFO or read from it for the last 4 Char times. | Reading from the FIFO (Receiver Buffer Register). |
| 0 | 0 | 1 | 4th | Transmitter holding register empty | Transmitter Holding Register Empty. | Writing to the Transmitter Holding Register or reading the IIR or LSR. |
| 0 | 0 | 0 | 5th | Modem status | nCTS, nDSR, nRI or nDCD. | Reading the Modem status register. |

UART\_FCR: The FIFO control register allows selection of the FIFO trigger level (the number of bytes that FIFO requires before enabling the Received Data Available interrupt). In addition, the FIFOs can be cleared using this register

Address: Operational Base + offset (0x0008)

Table 713.UART\_FCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7:6 | W | 0x0 | Define the Receiver FIFO Interrupt trigger level.  0x0 – 1 byte  0x1 – 4 bytes  0x2 – 8 bytes  0x3 – 14 bytes |
| 5:4 | - | 0x0 | Reserved |
| 3 | W | 0x0 | Hardware request to DMA.  ‘0’ – Disable.  ‘1’ – Enable. |
| 2 | W | 0x0 | Writing a ‘1’ to bit 2 clears the Transmitter FIFO and resets its logic. However, the shift register is not cleared, i.e., transmitting of the current character continues. |
| 1 | W | 0x0 | Writing a ‘1’ to bit 1 clears the Receiver FIFO and resets its logic. However, the shift register is not cleared, i.e., receiving of the current character continues. |
| 0 | W | 0x0 | Reserved. |

UART\_LCR: The line control register allows the specification of the format of the asynchronous data communication used. A bit in this register also allows access to the Divisor Latches, which define the baud rate. Reading from this register is allowed to check the current settings of the communication

Address: Operational Base + offset (0x000C)

Table 714.UART\_LCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7 | RW | 0x0 | Divisor Latch Access bit.  ‘1’ – The divisor latches can be accessed.  ‘0’ – The normal registers are accessed. |
| 6 | RW | 0x0 | Break Control bit.  ‘1’ – The serial out is forced into logic ‘0’ (break state).  ‘0’ – Break is disabled. |
| 5 | RW | 0x0 | Stick Parity bit.  ‘0’ – Stick Parity disabled.  ‘1’ - If bits 3 and 4 are logic ‘1’, the parity bit is transmitted and checked as logic ‘0’. If bit 3 is ‘1’ and bit 4 is ‘0’, then the parity bit is transmitted and checked as ‘1’. |
| 4 | RW | 0x0 | Even Parity select.  ‘0’ – Odd number of ‘1’ is transmitted and checked in  each word (data and parity combined). In other  words, if the data has an even number of ‘1’ in it,  then the parity bit is ‘1’.  ‘1’ – Even number of ‘1’ is transmitted in each word. |
| 3 | RW | 0x0 | Parity Enable.  ‘0’ – No parity.  ‘1’ – Parity bit is generated on each outgoing  character and is checked on each incoming  character. |
| 2 | RW | 0x0 | Specify the number of generated stop bits.  ‘0’ – 1 stop bit.  ‘1’ – 1.5 stop bits when a 5-bit character length  is selected, and 2 bits otherwise  Note that the receiver only checks the first stop bit. |
| 1:0 | RW | 0x3 | Number of bits in each character.  0x0 – 5 bits.  0x1 – 6 bits.  0x2 – 7 bits.  0x3 – 8 bits. |

UART\_MCR: The modem control register makes it possible to transfer control signals to a modem connected to the UART

Address: Operational Base + offset (0x0010)

Table 715.UART\_MCR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:5 | - | - | Reserved. |
| 4 | RW | 0x0 | Loopback mode.  ‘0’ – Normal operation.  ‘1’ – Loopback mode. When in loopback mode, the Serial Output Signal (TXD) is set to logic ‘1’. The signal of the transmitter shift register is internally connected to the input of the receiver shift register.  The following connections are made:  nDTR 🡺 nDSR nRTS 🡺 nCTS  Out1 🡺 nRI Out2 🡺 nDCD |
| 3 | RW | 0x0 | Out2. In loopback mode, connected to Data Carrier Detect (nDCD) input. |
| 2 | RW | 0x0 | Out1. In loopback mode, connected Ring Indicator (nRI) signal input. |
| 1 | RW | 0x0 | Request To Send (nRTS) signal control.  ‘0’ – nRTS is ‘1’ ‘1’ – nRTS is ‘0’ |
| 0 | RW | 0x0 | Data Terminal Ready (nDTR) signal control.  ‘0’ – nDTR is ‘1’ ‘1’ – nDTR is ‘0’ |

UART\_LSR: Line status register

Address: Operational Base + offset (0x0014)

Table 716.UART\_LSR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7 | R | 0x0 | ‘1’ – At least one parity error, framing error, or break indication has been received and is inside the FIFO. The bit is cleared upon reading from the register.  ‘0’ – Otherwise. |
| 6 | R | 0x1 | Transmitter Empty indicator.  ‘1’ – Both the transmitter FIFO and transmitter shift register are empty. The bit is cleared upon reading from the register or upon writing data to the transmit FIFO.  ‘0’ – Otherwise |
| 5 | R | 0x1 | Transmit FIFO is empty.  ‘1’ – The transmitter FIFO is empty. This generates Transmitter Holding Register Empty interrupt. The bit is cleared in the following cases: The LSR has been read, the IIR has been read, or data has been written to the transmitter FIFO.  ‘0’ – Otherwise. |
| 4 | R | 0x0 | Break Interrupt (BI) indicator.  ‘1’ – A break condition has been reached in the current character. The break occurs when the line is held at logic 0 for a time of one character (start bit + data + parity + stop bit). In this case, one zero character enters the FIFO and the UART waits for a valid start bit to receive the next character. The bit is cleared upon reading from the register. This generates a Receiver Line Status interrupt.  ‘0’ – No break condition in the current character. |
| 3 | R | 0x0 | Framing Error (FE) indicator.  ‘1’ – The received character at the top of the FIFO did not have a valid stop bit. The UART core tries re-synchronizing by assuming that the bit received was a start bit. It might be that all the following data is corrupt. The bit is cleared upon reading from the register. This generates a Receiver Line Status interrupt.  ‘0’ – No framing error in the current character |
| 2 | R | 0x0 | Parity Error (PE) indicator.  ‘1’ – The character that is currently at the top of the FIFO has been received with parity error. The bit is cleared upon reading from the register. This generates a Receiver Line Status interrupt.  ‘0’ – No parity error in the current character |
| 1 | R | 0x0 | Overrun Error (OE) indicator.  ‘1’ – The FIFO is full and another character has been received in the receiver shift register. If another character is starting to arrive, it will overwrite the data in the shift register, though the FIFO will remain intact. The bit is cleared upon reading from the register. This generates a Receiver Line Status interrupt.  ‘0’ – No overrun state. |
| 0 | R | 0x0 | Data Ready (DR) indicator.  ‘0’ – No characters in the FIFO.  ‘1’ – At least one character has been received and is in the FIFO. Reset when all data has been read out from FIFO. |

UART\_MSR: The register displays the current state of the modem control lines. Four bits also provide an indication of the state of one of the modem status lines. These bits are set to ‘1’ when a change in the corresponding line has been detected, and they are reset when the register is being read

Address: Operational Base + offset (0x0018)

Table 717.UART\_MSR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7 | R | 0x0 | Complement of the nDCD input or equal to Out2 (MCR[3]) in loopback mode. |
| 6 | R | 0x0 | Complement of the nRI input or equal to Out1 (MCR[2]) in loopback mode. |
| 5 | R | 0x0 | Complement of the nDSR input or equal to nDTR (MCR[0]) in loopback mode. |
| 4 | R | 0x0 | Complement of the nCTS input or equal to nRTS (MCR[1]) in loopback mode. |
| 3 | R | 0x0 | Delta Data Carrier Detect (DDCD) indicator  ‘1’ – The nDCD line has changed its state. |
| 2 | R | 0x0 | Delta Ring Indicator (DRI) detector  ‘1’ – The nRI line has changed its state.  Note: This definition is not the same as the Trailing Edge of Ring Indicator (TERI) detector. |
| 1 | R | 0x0 | Delta Data Set Ready (DDSR) indicator  ‘1’ – The nDSR line has changed its state. |
| 0 | R | 0x0 | Delta Clear To Send (DCTS) indicator  ‘1’ – The nCTS line has changed its state. |

UART\_ICR: IrDA control register

Address: Operational Base + offset (0x001C)

Table 718.UART\_ICR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:2 | - | - | Reserved. |
| 1 | RW | 0x0 | Sir enable.  ‘1’ - enable ‘0’- disable |
| 0 | RW | 0x0 | Sir loopback.  ‘1’ - enable ‘0’- disable |

UART\_CTRLR: UART baud rate clock source selection

Address: Operational Base + offset (0x0020)

Table 719.UART\_CTRLR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:1 | - | - | Reserved. |
| 0 | RW | 0x0 | Clock source select.  Reserve. |

UART\_DMA\_MODE: UART DMA mode select

Address: Operational Base + offset (0x0024)

Table 720.UART\_MODE

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:3 | - | - | Reserved. |
| 2 | RW | 0x0 | Tx : 0 single type ; 1 burst type. |
| 1 | RW | 0x0 | Rx : 0 single type ; 1 burst type. |
| 0 | RW | 0x0 | 0 socle dma ; 1 ARM dma. |

UART\_DLL/UART\_DLH

Setting the 7th bit of the LCR to ‘1’ can access the divisor latches. This bit should be restored to ‘0’ after setting the divisor latches to allow access to other registers that occupy the same addresses. The 2 bytes form one 16-bit register, which is internally accessed as a single number. You should therefore set all 2 bytes of the register to ensure normal operation. The register is set to the default value of 0 during reset. This disables all serial I/O operations and ensures explicit register setup in the software. The value set should be equal to (system clock speed) / (16 x desired baud rate).

The internal counter starts to work when the LSB of DL is written, so when setting the divisor, write the MSB first and the LSB last.

UART\_DLL: Divisor Latch LSB

Address: Operational Base + offset (0x0000)

Table 721.UART\_DLL

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7:0 | RW | 0x0 | The LSB of the baud-rate divisor latch. |

UART\_DLH: Divisor Latch MSB

Address: Operational Base + offset (0x0004)

Table 722.UART\_DLH

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved. |
| 7:0 | RW | 0x0 | The MSB of the baud-rate divisor latch. |

* 1. Programming Sequence

Control data is written to the UART line control register, UART\_LCR.

**UART\_LCR defines:**

* Transmission parameters
* Word length
* Number of transmission stop bits
* Parity mode
* Break generation

UART\_DLL and UART\_DLH define the baud rate divisor.

X generates an internal clock enable signal with a stream of one–UCLK-wide pulses and an average frequency of 16 times the desired baud rate. This signal is then divided by 16 to yield the transmit clock.

**Data Transmission or Reception**

Data received or transmitted is stored in two 16-byte FIFOs. For transmission, data is first written into the transmit FIFO. If the UART is enabled, it causes a data frame to start transmitting with the parameters indicated in the UART\_LCR. Data continues to be transmitted until there is no data left in the transmit FIFO. This produces 16 samples of the UART bit. Three readings are taken for each bit sample, and the majority value is kept. The following paragraphs define the middle sampling point and one sample is taken from either side of it.

When the receiver is idle (RXD continuously 1) and a LOW is detected on the data input (a start bit has been received), the receive counter, with the clock enabled by Baud16, begins running. Data is sampled on the eighth cycle of that counter in normal UART mode, or the fourth cycle of the counter in SIR mode to allow for the shorter logic 0 pulses (half way through a bit period).

The start bit is valid if RXD is still LOW on the eighth cycle of the sample clock. Otherwise, a false start bit is detected and it is ignored.

If the start bit was valid, successive data bits are sampled on every 16th cycle of the sample clock (one bit period later) according to the programmed length of the data characters. The parity bit is then checked to determine if parity mode is enabled.

Finally, a valid stop bit is confirmed if RXD is HIGH, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO.

**Overrun Condition**

The overrun error is set when the FIFO is full and the next character is completely received in the shift register. In this case, the data in the shift register is overwritten, but it is not written into the FIFO. When an empty location is available in the receive FIFO, and another character is received, the received character is copied into the receive FIFO. The overrun state is then cleared.

**UART Packet Frame**



* Compliance with AMBA APB 2.0 protocol
* Support for up to a 115.2Kbps baud-rate
* Separate transmit and receive FIFO buffers (16 x 8) to reduce CPU interrupts
* Programmable baud rate generator. This enables the division of the internal clock by (1 ~ 65535 x 16) and generates an internal x16 clock
* Standard asynchronous communication bits (start, stop, and parity). These are added prior to transmission and removed on reception
* Independent masking of transmit FIFO, receive FIFO, and receive timeout and error condition interrupts
* False start bit detection
* Line break generation and detection
* Fully-programmable serial interface characteristics:
* Data can be 5, 6, 7, 8 bits
* Even, odd, stick, or no-parity bit generation and detection
* 1, 1.5, or 2-stop bit generation

**Baud rate generation**

Figure 57.Transmit programming

* 1. Programming Restrictions

No programming restriction

1. Watch Timer
   1. Register Summary

This chapter describes the control/status registers of the design.

Software should read and write these registers using 32-bits accesses.

Table 723.Watch Timer Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| WDTLR | 0x0000 | 0x00000000 | Watchdog load register |
| WDTCVR | 0x0004 | 0x0000FFFF | Watchdog current value register |
| WDTCON | 0x0008 | 0x00000000 | Watchdog control register |
| TEST | 0x0010 | 0x00000000 | Test mode register |

* 1. Register Descriptions

WDTLR :Load register for timer counter

Address: Operational Base + offset (0x0000)

Table 724.WDTLR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | WO | 0x0 | Load register for timer counter.Range from 232-1 |

WDTCVR: Current value register

Address: Operational Base + offset (0x0004)

Table 725.WDTCVR

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RO | 0xFFFF | The current counter value |

WDTCON Control register

Address: Operational Base + offset (0x00008)

Table 726.WDTCON

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:5 | RW | - | Reserved |
| 4 | RW | 0x0 | Reset output enable/disable. Enables or disables the reset signal bit of the watchdog timer output.  0: Disables the reset function of the watchdog timer.  1: Enables the reset signal of the μplatform core at watchdog timeout. |
| 3 | RW | 0x0 | Watchdog Timer enable/disable.  0: Disable 1: Enable |
| 2:0 | RW | 0x0 | Prescale factor.  0x0: 1 0x1: 1/4  0x2: 1/8 0x3: 1/16  0x4: 1/32 0x5: 1/64  0x6: 1/128 0x7: 1/256 |

TEST: Test register

Address: Operational Base + offset (0x0010)

Table 727.TEST

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:1 | WO | - | Reserved |
| 0 | WO | 0x0 | Test mode, the bit is set ‘1’ , test the prescale from 0xff to 0x00. |

* 1. Programming Sequence

The flow chart below describes how the software configure and performs a watchdog working.

begin

config WDTCON

TEST=1?

config TEST

WDTCON=32’h18

wdtrst\_n=1

end

yes

no

WDTCON[4]=1?

WDTCON[3]=1?

config WDTLR

reset enable

counter timer enable

counter timer disable

reset disable

end

end

end

end

no

no

yes

yes

Figure 58.WDT control flow chart

* 1. Programming Restrictions

No programming restriction

1. Timer
   1. Register Summary

Table 728.Timer Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| TMR0LR | 0x0000 | 0x00000000 | Load register. |
| TMR0CVR | 0x0004 | 0x00000000 | Current value register. |
| TMR0CON | 0x0008 | 0x00000002 | Control register. |
| TMR1LR | 0x0010 | 0x00000000 | Load register. |
| TMR1CVR | 0x0014 | 0x00000000 | Current value register. |
| TMR1CON | 0x0018 | 0x00000002 | Control register. |
| TMR2LR | 0x0020 | 0x00000000 | Load register. |
| TMR2CVR | 0x0024 | 0x00000000 | Current value register. |
| TMR2CON | 0x0028 | 0x00000002 | Control register |
| TMRMODE | 0x0030 | 0x00000000 | Mode Select |

* 1. Register Descriptions

TMRxLR(x=0,1,2): Load register for the timer counter

Address: Operational Base + offset (0x0000, 0x0010, 0x0020)

Table 729.TMRxLR(x=0,1,2)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | W | 0x0 | Load register for the timer counter. Counting value ranges from 0 ~ (232 –1). |

TMRxCVR(x=0,1,2): Current value register

**Address: Operational Base + offset (0x0004, 0x0014, 0x0024)**

Table 730.TMRxCVR(x=0,1,2)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | R | 0x0 | The current counter value. |

TMRxCON(x=0,1,2):Control register

Address: Operational Base + offset (0x0008, 0x0018, 0x0028)

Table 731.TMRxCON(x=0,1,2)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:11 | - | - | Reserved |
| 10 | RW | 0x0 | Counter enable/disable  0: Enable 1: Disable |
| 9 | RW | 0x0 | Reserved. |
| 8 | RW | 0x0 | Timer enable/disable.  0: Disable 1: Enable |
| 7 | RW | 0x0 | Timer counting mode selection.  0: Free-Running 1: Periodical |
| 6:4 | RW | 0x0 | Prescale factor.  0x0: 1 0x1: 1/4  0x2: 1/8 0x3: 1/16  0x4: 1/32 0x5: 1/64  0x6: 1/128 0x7: 1/256 |
| 3 | RW | 0x0 | Interrupt mask.  0: Disable 1: Enable |
| 2 | RW | 0x0 | Interrupt clear.  This bit is set when an interrupt is pending. Writing a ‘0’ to this bit will clear the interrupt |
| 1 | RW | 0x1 | Interrupt trigger type.  0: Edge trigger 1: Level trigger |
| 0 | - | - | Reserved. |

TMRMODE: Mode Select register

Address: Operational Base + offset (0x0030)

Table 732.TMRMODE

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Contains initial value of counter.  00: normal function.  01: TMR0\_INT always block ,just working in edge trigger.  10: TMR0\_INT and TMR1\_INT always block , just working in edge trigger. |

* 1. Programming Sequence

The timer is loaded by writing to the TMRxLR and, if enabled, counting down to zero. An interrupt is generated upon reaching zero. Writing to the interrupt control bit of the TMRxCON can clear the interrupt.

If the timer is operating in free-running mode after reaching a zero count, it continues to decrement from its maximum value (232 - 1). If periodical mode is selected, the timer reloads the count value from the TMRxLR and continues to decrement. Under this mode, the counter effectively generates a periodical interrupt. The mode is selected by setting bit 6 of the TMRxCON.

At any point, the current counter value may be read from the TMRxCVR via the APB interface.

The entire scenario can be explained by the figure below:



A pre-scale unit generates the timer clock. The enable is then used by the counter to create a clock with a timing of one of the following:

* PCLK
* PCLK/4, PCLK/8, PCLK/16, PCLK/32, PCLK/64, PCLK/128, PCLK/256

The pre-scale value can be determined by programming the bits 6-4 in the TMRxCON register.

The counter clock frequency can be evaluated using the formula given below:

F timer = F PCLK / (Pre-scale \* Counter Value)

For example:

Ex1 : PCLK is 100MHz, set Counter Value as 100M, Pre-scale value as 1,

Ttimer = 1 us \* 100 M \* 1 = 1 second.

Ex2 : PCLK is 100MHz, set Counter Value as 100M, Pre-scale value as 5,

Ttimer = 1 us \* 100 M \* 5 = 5 second.

* 1. Programming Restrictions

No programming restriction

1. GPIO Controller
   1. Register Summary

Table 733.GPIO Controller Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| GPIO\_PADR | 0x0000 | 0x00000000 | Port A data register. |
| GPIO\_PACON | 0x0004 | 0x00000000 | Port A direction register. |
| GPIO\_PBDR | 0x0008 | 0x00000000 | Port B data register. |
| GPIO\_PBCON | 0x000C | 0x00000000 | Port B direction register. |
| GPIO\_PCDR | 0x0010 | 0x00000000 | Port C data register. |
| GPIO\_PCCON | 0x0014 | 0x00000000 | Port C direction register. |
| GPIO\_PDDR | 0x0018 | 0x00000000 | Port D data register. |
| GPIO\_PDCON | 0x001C | 0x00000000 | Port D direction register. |
| GPIO\_TEST | 0x0020 | 0x00000000 | GPIO function test register. |
| GPIO\_IEA | 0x0024 | 0x00000000 | Port A interrupt mask register. |
| GPIO\_IEB | 0x0028 | 0x00000000 | Port B interrupt mask register. |
| GPIO\_IEC | 0x002C | 0x00000000 | Port C interrupt mask register. |
| GPIO\_IED | 0x0030 | 0x00000000 | Port D interrupt mask register. |
| GPIO\_ISA | 0x0034 | 0x00000000 | Port A interrupt sense register. |
| GPIO\_ISB | 0x0038 | 0x00000000 | Port B interrupt sense register. |
| GPIO\_ISC | 0x003C | 0x00000000 | Port C interrupt sense register. |
| GPIO\_ISD | 0x0040 | 0x00000000 | Port D interrupt sense register. |
| GPIO\_IBEA | 0x0044 | 0x00000000 | Port A interrupt both-edges register. |
| GPIO\_IBEB | 0x0048 | 0x00000000 | Port B interrupt both-edges register. |
| GPIO\_IBEC | 0x004C | 0x00000000 | Port C interrupt both-edges register. |
| GPIO\_IBED | 0x0050 | 0x00000000 | Port D interrupt both-edges register. |
| GPIO\_IEVA | 0x0054 | 0x00000000 | Port A interrupt event register. |
| GPIO\_IEVB | 0x0058 | 0x00000000 | Port B interrupt event register. |
| GPIO\_IEVC | 0x005C | 0x00000000 | Port C interrupt event register. |
| GPIO\_IEVD | 0x0060 | 0x00000000 | Port D interrupt event register. |
| GPIO\_ICA | 0x0064 | 0x00000000 | Port A interrupt clear register. |
| GPIO\_ICB | 0x0068 | 0x00000000 | Port B interrupt clear register. |
| GPIO\_ICC | 0x006C | 0x00000000 | Port C interrupt clear register. |
| GPIO\_ICD | 0x0070 | 0x00000000 | Port D interrupt clear register. |
| GPIO\_ISR | 0x0074 | 0x00000000 | GPIO interrupt status register. |

* 1. Register Descriptions

GPIO\_PxDR(x=A, B, C, D): The GPIO\_PxDR is the Port x data register

Address: Operational Base + offset (0x00/0x08/0x10/0x18)

Table 734.GPIO\_PxDR(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | Port x data register. Write this address and Read this address may not be the same value. |

GPIO\_PxCON(x=A, B,C,D): The GPIO\_PxCON is the Port x data direction register. When asserted HIGH, the corresponding pin is configured as an output. When asserted LOW, it is configured as an input. At reset, all bits are configured as inputs.

Address: Operational Base + offset (0x04/0x0C/0x14/0x1c)

Table 735.GPIO\_PxCON(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | Port x direction register.  0: Input ,open the debounce module  1: Output,open the debounce module |

GPIO\_TEST: GPIO function test register

Address: Operational Base + offset (0x20)

Table 736.GPIO\_TEST

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:3 | - | - | Reserved |
| 2:1 | RW | 0x0 | Test mode.  0x0 – PORTB -> PORTA  0x1 – PORTA -> PORTB  0x2 – PORTD -> PORTC.  0x3 – PORTC -> PORTD. |
| 0 | RW | 0x0 | Test mode enable indicator.  ‘1’ – The GPIO loop-back test mode enable.  ‘0’ – Normal mode. |

GPIO\_IEx (x=A, B, C, D): The GPIO\_IEx register is the Port x interrupt mask register. Bits set to HIGH in GPIO\_IEx allow the corresponding pins to trigger their individual interrupts and the combined gpio\_int line. On the other hand, bits set to LOW represent disable interrupt triggers on the pins. The default value is set to LOW at reset

Address: Operational Base + offset (0x24/0x28/0x2C/0x30)

Table 737.GPIO\_IEx(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:3 | - | - | Reserved |
| 7:0 | RW | 0x0 | Port x interrupt mask register.  0: Masked 1: Not masked. |

GPIO\_ISx (x=A, B, C, D): The GPIO\_ ISx register is the Port x interrupt sense register. The default bit setting is edge sensitive upon reset

Address: Operational Base + offset (0x34/0x38/0x3c/0x40)

Table 738.GPIO\_ISx(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | Port x interrupt sense register.  0: Edge on corresponding pin is detected.  1: Level on corresponding pin is detected. |

GPIO\_IBEx (x=A, B, C, D): The GPIO\_ IBEx register is the Port x interrupt both-edges register. When the bits in GPIO\_ISx are set to edge sensitive, bits set to HIGH in GPIO\_IBEx configure the corresponding pin to detect both rising and falling edges, regardless of the corresponding bit in the GPIO\_IEVx. The default value is set to LOW upon reset

Address: Operational Base + offset (0x44/0x48/0x4c/0x50)

Table 739.GPIO\_IBEx(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | Port x interrupt both-edges register.  0: Single edge, interrupt generation event is controlled by GPIO\_IEVx.  1: Both edges on a corresponding pin trigger an interrupt. |

GPIO\_IEVx (x=A, B, C, D): The GPIO\_ IEVx register is the Port x interrupt event register. When the corresponding bit is asserted to HIGH in GPIO\_IEVx, it is configured to detect rising edges or high levels, depending on the corresponding bit value in GPIO\_ISx. Clearing a bit configures the pin to detect falling edges or low levels, depending on the corresponding bit value in GPIO\_ISx. The default value is set to LOW upon reset

Address: Operational Base + offset (0x54/0x58/0x5c/0x60)

Table 740.GPIO\_IEVx(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | RW | 0x0 | Port x interrupt event register.  0: Falling edges or low levels on corresponding pins trigger interrupts.  1: Rising edges or high levels on corresponding pins trigger interrupts. |

GPIO\_ICx (x=A, B, C, D): The GPIO\_ ICx register is the Port x interrupt clear register. Writing a 1 to a bit in this register clears the corresponding interrupt edge detection logic register. Writing a 0 has no effect. The register is write-only and all bits are cleared by a reset

Address: Operational Base + offset (0x64/0x68/0x6c/0x70)

Table 741.GPIO\_ICx(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7:0 | W | 0x0 | Port x interrupt clear register.  0: No effect.  1: Clears the corresponding interrupt detection logic register. |

GPIO\_ISR: The GPIO\_ ISR register is the interrupt status register. Bits read HIGH in this register indicate that interrupt trigger conditions have been detected. Bits read LOW indicate that corresponding input pins have not initiated an interrupt. This register is read-only and all bits are cleared by a reset

Address: Operational Base + offset (0x0074)

Table 742.GPIO\_ISRx(x=A,B,C,D)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:24 | R | 0x0 | Port D interrupt status register.  The status of the interrupt trigger condition detection on pins.  0: GPIO interrupt not active.  1: GPIO asserting interrupt. |
| 23:16 | R | 0x0 | Port C interrupt status register.  The status of the interrupt trigger condition detection on pins.  0: GPIO interrupt not active.  1: GPIO asserting interrupt. |
| 15:8 | R | 0x0 | Port B interrupt status register.  The status of the interrupt trigger condition detection on pins.  0: GPIO interrupt not active.  1: GPIO asserting interrupt. |
| 7:0 | R | 0x0 | Port A interrupt status register.  The status of the interrupt trigger condition detection on pins.  0: GPIO interrupt not active.  1: GPIO asserting interrupt. |

* 1. Programming Sequence
     1. Rising Edge Detected

1. Bits set to “0” in GPIO\_ISx detect an edge on the corresponding pin.
2. Bits set to “0” in GPIO\_IBEx configure the corresponding pin to detect a single edge.
3. Bits set to “1” in GPIO\_IEVx configure the corresponding pin to detect a rising edge.
4. Bits set to “1” in GPIO\_IEx allow the corresponding pins to trigger their individual interrupts.
5. When one or more input data pins cause a rising edge-triggered interrupt, interrupt output gpio\_int can be set to “1”.

FSM

Table 743.Rising edge detected timing

Note:gpio\_int\_temp is taking the edge of px\_din[0] to generate the pluse.

* + 1. Falling Edge Detected

1. Bits set to “0” in GPIO\_ISx detect an edge on the corresponding pin.
2. Bits set to “0” in GPIO\_IBEx configure the corresponding pin to detect a single edge.
3. Bits set to “0” in GPIO\_IEVx configure the corresponding pin to detect a falling edge.
4. Bits set to “1” in GPIO\_IEx allow the corresponding pins to trigger their individual interrupts.
5. When one or more input data pins cause a falling edge-triggered interrupt, interrupt output gpio\_int can be set to “1”.

FSM

Figure 59.Falling edge detected timing

Note:gpio\_int\_temp is taking the edge of px\_din[0] to generate the pluse.

* + 1. Both Edges Detected

1. Bits set to “0” in GPIO\_ISx detect an edge on the corresponding pin.
2. Bits set to “1” in GPIO\_IBEx configure the corresponding pin to detect both edges.
3. Bits set to “0” or “1” in GPIO\_IEVx are okay.
4. Bits set to “1” in GPIO\_IEx allow the corresponding pins to trigger their individual interrupts.
5. When one or more input data pins cause a falling edge-triggered or rising edge-triggered interrupt, interrupt output gpio\_int can be set to “1”.

FSM

Figure 60.Both-edges detected timing

Note:gpio\_int\_temp is taking the edge of px\_din[0] to generate the pluse.

* + 1. High Level Detected

1. Bits set to “1” in GPIO\_ISx detect the level of the corresponding pin.
2. Bits set to “0” or “1” in GPIO\_IBEx are okay.
3. Bits set to “1” in GPIO\_IEVx configure the corresponding pin to detect a high level.
4. Bits set to “1” in GPIO\_IEx allow the corresponding pins to trigger their individual interrupts.
5. When one or more input data pins causes a high-level interrupt, interrupt output gpio\_int can be set to “1”.

FSM

Figure 61.High-level detected timing

Note:GPIO\_ICx can clear the gpio\_int . But if the level detected trigger conditions still hold , then GPIO\_ICx only clear a cycle interrupt , so sw is hard to see its function .

* + 1. Low Level Detected

1. Bits set to “1” in GPIO\_ISx detect the level of the corresponding pin.
2. Bits set to “0” or “1” in GPIO\_IBEx are okay.
3. Bits set to “0” in GPIO\_IEVx configure the corresponding pin to detect a low level.
4. Bits set to “1” in GPIO\_IEx allow the corresponding pins to trigger their individual interrupts.
5. When one or more input data pins causes a low level interrupt, interrupt output gpio\_int can be set to “1”.

FSM

Figure 62.High-level detected timing

Note:GPIO\_ICx can clear the gpio\_int . But if the level detected trigger conditions still hold , then GPIO\_ICx only clear a cycle interrupt , so sw is hard to see its function .

* 1. Programming Restrictions

No programming restriction

1. PWM
   1. Register Summary

Table 744.PWMRegister summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| PWMT0\_CNTR | 0x0000 | 0x00000000 | Main counter register |
| PWMT0\_HRC | 0x0004 | 0x00000000 | PWM HIGH Reference/Capture register |
| PWMT0\_LRC | 0x0008 | 0x00000000 | PWM LOW Reference/Capture register |
| PWMT0\_CTRL | 0x000C | 0x00000000 | PWM control register |
| PWMT1\_CNTR | 0x0010 | 0x00000000 | Main counter register |
| PWMT1\_HRC | 0x0014 | 0x00000000 | PWM HIGH Reference/Capture register |
| PWMT1\_LRC | 0x0018 | 0x00000000 | PWM LOW Reference/Capture registe |
| PWMT1\_CTRL | 0x001C | 0x00000000 | PWM control register |
| PWMT2\_CNTR | 0x0020 | 0x00000000 | Main counter register |
| PWMT2\_HRC | 0x0024 | 0x00000000 | PWM HIGH Reference/Capture register |
| PWMT2\_LRC | 0x0028 | 0x00000000 | PWM LOW Reference/Capture register |
| PWMT2\_CTRL | 0x002C | 0x00000000 | PWM control register |
| PWMT3\_CNTR | 0x0030 | 0x00000000 | Main counter register |
| PWMT3\_HRC | 0x0034 | 0x00000000 | PWM HIGH Reference/Capture register |
| PWMT3\_LRC | 0x0038 | 0x00000000 | PWM LOW Reference/Capture register |
| PWMT3\_CTRL | 0x003C | 0x00000000 | PWM control register |

* 1. Register Descriptions

PWMTx\_CNTR (x=0~3): PWM0 timer counter

Address: Operational Base + offset (0x0000/0x0010/0x0020/0x0030)

Table 745.PWMTx\_CNTR(x=0~3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0 | Main PWM timer counter. Counting value ranges from 0 ~ (232 –1).  Note: the value must be less than LRC |

PWMTx\_HRC (x=0~3): PWM0 HIGH reference or capture register

Address: Operational Base + offset (0x0004/0x0014/0x0024/0x0034)

Table 746.PWMTx\_HRC(x=0~3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0 | PWM HIGH reference/capture registers |

PWMTx\_LRC (x=0~3): PWM0 LOW reference or capture register

Address: Operational Base + offset (0x0008/0x0018/0x0028/0x0038)

Table 747.PWMTx\_LRC(x=0~3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:0 | RW | 0 | PWM LOW reference/capture registers |

PWMTx\_CTRL (x=0~3) :Control register of the PWM0 Timer

Address: Operational Base + offset (0x000C/0x001C/0x002C/0x003C)

Table 748.PWMTx\_CTRL(x=0~3)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Type | Reset Value | Description |
| 31:14 | - | - | Reserved |
| 13:9 | RW | 0 | Prescale factor.  The counter clock is divided by the prescale factor.  Counter clock = PCLK/2PWMx\_CTRL[13:9]+1 |
| 8 | RW | 0 | Capture mode enable/disable  0: Disable 1: Enable |
| 7 | RW | 0 | PWM reset.  0: Normal operation 1: Reset PWM |
| 6 | RW | 0 | Interrupt status and clear bit.  Write “1” to clear interrupt status. |
| 5 | RW | 0 | PWM timer interrupt enable/disable.  The PWM timer will assert an interrupt when the PWMTx\_CNTR value is equal to the value of PWMTx\_LRC or PWMTx\_HRC.  0: Disable 1: Enable |
| 4 | RW | 0 | Single counter mode.  0: PWMTx\_CNTR is restarted after it reaches a value equal to the PWMTx\_LRC value.  1: PWMTx\_CNTR is not increased anymore after it reaches a value equal to the PWMT0\_LRC value. |
| 3 | RW | 0 | PWM output enable/disable.  0: Disable 1: Enable |
| 2:1 | - | - | Reserved. |
| 0 | RW | 0 | PWM timer enable/disable.  0: Disable 1: Enable |

* 1. Programming Sequence

The PWM timer is the Pulse-width modulation that can be programmed by controlling the PWMx\_CTRL registers. To operate the PWM Timer, users should set the PWMx\_HRC and the PWMx\_LRC with the value of the low and high periods of the PWM Timer output data, respectively. PWMx\_HRC will be the number of clock cycles after the reset of the PWMx\_CNTR, when PWM output should go high. In addition, PWMx\_LRC will be the number of clock cycles after reset of the PWMx\_CNTR, when PWM output should go low. PWMx\_CNTR can be reset with the hardware reset.

Pre-Scale function

PWM Time uses a timing pre-scale function to divide the PCLK clock source before operating the PWM Timer. A pre-scale unit generates the timer clock. The count then uses the enable to create a clock with a timing of one of the following:

**Counter Clock = PCLK/2PWMx\_CTRL[13:9]+1**

The pre-scale value can be determined by programming bits 13 ~ bit9 in the PWMTx\_CTRL registers.

Below description is former limitation, already modify in 8028 design. We can change the pre-scale value during operation.

For PWM stabilizati**o**n, do not change the pre-scale value during the system operation. Further, reset the system before changing the pre-scale value.

Interrupt Feature

Whenever the value of the PWMx\_HRC or PWMx\_LRC equals the value of the PWMx\_CNTR, an interrupt request can be asserted. Bit 6 in PWMTx\_CTRL registers can be set to clear the interrupt status.

* 1. Programming Restrictions

No programming restriction

1. IOMUX
   1. Programming Terminology

Table 1. Programming terminology

|  |  |
| --- | --- |
| Name | Description |
| RW | Read &Write |

* 1. Register Summary

Table 2. Register summary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Offset | Reset Value | Description |
| Main Clock Pad Control Register | | | |
| PAD\_CTRL\_0 | 0x000 | 0x0 | [31:4]: Reserved  [3:0]:X\_MAIN\_CLK |
| SDIO Pad Control Register | | | |
| PAD\_CTRL\_1 | 0x004 | 0x0 | [31:24]: X\_SDIO0\_SDCLK  [23:16]: X\_SDIO0\_CMD  [15:8]: X\_SDIO0\_DAT \_0  [7:0]: X\_SDIO0\_DAT \_1 |
| PAD\_CTRL\_2 | 0x008 | 0x0 | [31:24]: X\_SDIO0\_DAT \_2  [23:16]: X\_SDIO0\_DAT \_3  [15:8]: Reserved  [7:0]: Reserved |
| PAD\_CTRL\_3 | 0x00c | 0x0 | [31:24]: X\_SDIO1\_SDCLK  [23:16]: X\_SDIO1\_CMD  [15:8]: X\_SDIO1\_DAT \_0  [7:0]: X\_SDIO1\_DAT \_1 |
| PAD\_CTRL\_4 | 0x010 | 0x0 | [31:24]: X\_SDIO1\_DAT \_2  [23:16]: X\_SDIO1\_DAT \_3  [15:8]: Reserved  [7:0]: Reserved |
| GMAC Pad Control Register | | | |
| PAD\_CTRL\_5 | 0x014 | 0x0 | [31:24]: X\_GMAC\_CLKT  [23:16]: X\_GMAC\_CLKR  [15:8]: X\_GMAC\_PHY\_50M  [7:0]: X\_GMAC\_PHY\_125M |
| PAD\_CTRL\_6 | 0x018 | 0x0 | [31:24]: X\_GMAC\_CLKO  [23:16]: X\_GMAC\_TXD\_0  [15:8]: X\_GMAC\_TXD\_1  [7:0]: X\_GMAC\_TXD\_2 |
| PAD\_CTRL\_7 | 0x01c | 0x0 | [31:24]: X\_GMAC\_TXD\_3  [23:16]: X\_GMAC\_TXD\_4  [15:8]: X\_GMAC\_TXD\_5  [7:0]: X\_GMAC\_TXD\_6 |
| PAD\_CTRL\_8 | 0x020 | 0x0 | [31:24]: X\_GMAC\_TXD\_7  [23:16]: X\_GMAC\_MDIO  [15:8]: X\_GMAC\_MDC  [7:0]: X\_GMAC\_RXER |
| PAD\_CTRL\_9 | 0x024 | 0x0 | [31:24]: X\_GMAC\_RXDV  [23:16]: X\_GMAC\_COL  [15:8]: X\_GMAC\_CRS  [7:0]: X\_GMAC\_RXD\_0 |
| PAD\_CTRL\_10 | 0x028 | 0x0 | [31:24]: X\_GMAC\_RXD\_1  [23:16]: X\_GMAC\_RXD\_2  [15:8]: X\_GMAC\_RXD\_3  [7:0]: X\_GMAC\_RXD\_4 |
| PAD\_CTRL\_11 | 0x02c | 0x0 | [31:24]: X\_GMAC\_RXD\_5  [23:16]: X\_GMAC\_RXD\_6  [15:8]: X\_GMAC\_RXD\_7  [7:0]: X\_GMAC\_TXEN |
| PAD\_CTRL\_12 | 0x030 | 0x0 | [31:24]: Reserved  [23:16]: Reserved  [15:8]: Reserved  [7:0]: X\_GMAC\_TXER |
| Dedicated GPIO Pad Control Register | | | |
| PAD\_CTRL\_13 | 0x034 | 0x0 | [31:24]: Reserved  [23:16]: X\_SYSPWREN  [15:8]: X\_WAKE\_IRQ  [7:0]: X\_SYS\_RST\_N |
| PAD\_CTRL\_14 | 0x038 | 0x0 | [31:24]: X\_RTC\_SCL  [23:16]: X\_RTC\_SDA  [15:8]: X\_RTC\_INT  [7:0]: X\_RTC\_32K\_CLK |
| PAD\_CTRL\_15 | 0x03c | 0x0 | Reserved |
| PAD\_CTRL\_16 | 0x040 | 0x0 | Reserved |
| PAD\_CTRL\_17 | 0x044 | 0x0 | [31:24]: X\_IVA\_ EXTCLK  [23:16]: X\_IVA\_ PIXCLK  [15:8]: X\_IVA\_FRAME\_VALID  [7:0]: X\_IVA\_LINE\_VALID |
| PAD\_CTRL\_18 | 0x048 | 0x0 | [31:24]: X\_IVA\_DATA\_0  [23:16]: X\_IVA\_DATA\_1  [15:8]: X\_IVA\_DATA\_2  [7:0]: X\_IVA\_DATA\_3 |
| PAD\_CTRL\_19 | 0x04c | 0x0 | [31:24]: X\_IVA\_DATA\_4  [23:16]: X\_IVA\_DATA\_5  [15:8]: X\_IVA\_DATA\_6  [7:0]: X\_IVA\_DATA\_7 |
| PAD\_CTRL\_20 | 0x050 | 0x0 | [31:24]: X\_IVA\_DATA\_8  [23:16]: X\_IVA\_DATA\_9  [15:8]: X\_IVA\_DATA\_10  [7:0]: X\_IVA\_DATA\_11 |
| PAD\_CTRL\_21 | 0x054 | 0x0 | [31:24]: Reserved  [23:16]: Reserved  [15:8]: X\_SDIO0\_SDCD\_N  [7:0]: X\_SDIO0\_SDWP\_N |
| PAD\_CTRL\_22 | 0x058 | 0x0 | [31:24]: X\_SDIO0\_CLE  [23:16]: X\_SDIO0\_LED  [15:8]: X\_SDIO0\_BUS\_POW  [7:0]: X\_SDIO0\_BUS\_VOLT |
| Alternated GPIO Control Register | | | |
| PAD\_CTRL\_23 | 0x05c | 0x0 | [29:20]: X\_SDIO1\_SDCD\_N  [19:10]: X\_SDIO1\_SDWP\_N  [9:0]: X\_SDIO1\_CLE |
| PAD\_CTRL\_24 | 0x060 | 0x0 | [29:20]: X\_SDIO1\_LED  [19:10]: X\_SDIO1\_BUS\_POW  [9:0]: X\_SDIO1\_BUS\_VOLT |
| PAD\_CTRL\_25 | 0x064 | 0x0 | [29:20]:X\_PWM0  [19:10]:X\_PWM1  [9:0]:X\_PWM2 |
| PAD\_CTRL\_26 | 0x068 | 0x0 | [29:20]: X\_SPI0\_SCLK  [19:10]: X\_SPI0\_MOSI  [9:0]: X\_SPI0\_MISO |
| PAD\_CTRL\_27 | 0x06c | 0x0 | [29:20]: X\_SPI0\_SS\_N  [19:10]: X\_SPI1\_SCLK  [9:0]: X\_SPI1\_MOSI |
| PAD\_CTRL\_28 | 0x070 | 0x0 | [29:20]: Reserved  [19:10]: X\_SPI1\_MISO  [9:0]: X\_SPI1\_SS\_N |
| PAD\_CTRL\_29 | 0x074 | 0x0 | [29:20]: Reserved  [19:10]: X\_I2C\_SCL  [9:0]: X\_I2C\_SDA |
| PAD\_CTRL\_30 | 0x078 | 0x0 | [29:20]: Reserved  [19:10]: X\_UART\_TXD  [9:0]: X\_UART\_RXD |
| PAD\_CTRL\_31 | 0x07c | 0x0 | [29:20]: X\_GPIO\_PA\_0  [19:10]: X\_GPIO\_PA\_1  [9:0]: X\_GPIO\_PA\_2 |
| PAD\_CTRL\_32 | 0x080 | 0x0 | [29:20]: X\_GPIO\_PA\_3  [19:10]: X\_GPIO\_PA\_4  [9:0]: X\_GPIO\_PA\_5 |
| PAD\_CTRL\_33 | 0x084 | 0x0 | [29:20]: Reserved  [19:10]: X\_GPIO\_PA\_6  [9:0]: X\_GPIO\_PA\_7 |
| PAD\_CTRL\_34 | 0x088 | 0x0 | [29:20]: X\_GPIO\_PB\_0  [19:10]: X\_GPIO\_PB\_1  [9:0]: X\_GPIO\_PB\_2 |
| PAD\_CTRL\_35 | 0x08c | 0x0 | [29:20]: X\_GPIO\_PB\_3  [19:10]:X\_GPIO\_PB\_4  [9:0]:X\_GPIO\_PB\_5 |
| PAD\_CTRL\_36 | 0x090 | 0x0 | [29:20]: Reserved  [19:10]: X\_GPIO\_PB\_6  [9:0]:X\_GPIO\_PB\_7 |
| PAD\_CTRL\_37 | 0x094 | 0x0 | [29:20]: X\_GPIO\_PC\_0  [19:10]: X\_GPIO\_PC\_1  [9:0]: X\_GPIO\_PC\_2 |
| PAD\_CTRL\_38 | 0x098 | 0x0 | [29:20]: X\_GPIO\_PC\_3  [19:10]:X\_GPIO\_PC\_4  [9:0]:X\_GPIO\_PC\_5 |
| PAD\_CTRL\_39 | 0x09c | 0x0 | [29:20]: Reserved  [19:10]: X\_GPIO\_PC\_6  [9:0]:X\_GPIO\_PC\_7 |
| PAD\_CTRL\_40 | 0x0a0 | 0x0 | [29:20]: X\_SPI\_NOR\_SCLK  [19:10]: X\_SPI\_NOR\_MOSI  [9:0]: X\_SPI\_NOR\_MISO |
| PAD\_CTRL\_41 | 0x0a4 | 0x0 | [29:20]: X\_SPI\_NOR\_SS\_N  [19:10]: X\_SPI\_NOR\_SIO2\_IN  [9:0]:X\_SPI\_NOR\_SIO3\_IN |
| Engineering Mode Select | | | |
| SW\_MUX\_SEL | 0x100 | 0x0 | Software mode select register |
| DBG\_BANK\_SEL0 | 0x104 | 0x0 | IP debug pin bank select register |
| DBG\_BANK\_SEL1 | 0x108 | 0x0 | IP debug pin bank select register |
| Voltage Control Register | | | |
| PAD\_VOLT\_CTRL | 0x110 | 0x0 | Software PAD voltage control register |
| CoreSight Trace Select Register | | | |
| CS\_TRACE\_SEL | 0x114 | 0x0 | CoreSight trace function select register |

* 1. Register Descriptions
     1. Pad Control Register

**Main clock Pad Control Register**



Figure 1. Bit map for main-clock pad control register

* PAD\_CTRL\_0 Register (X\_MAIN\_CLK)

Address: Operational Base + offset (0x00)



Figure 2. PAD\_CTRL\_0 Register

Table 3. PAD\_CTRL\_0 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:4 | - | - | Reserved |
| X\_MAIN\_CLK | 3 | RW | 0x0 | Function enable signal, high active |
| 2 | RW | 0x0 | Active high oscillator enable signal |
| 1:0 | RW | 0x0 | E2, E1: Drive control pins  00:3mA, 01:6mA, 10:9mA, 11:13mA |

**SDIO Pad Control Register**



Figure 3. Bit map for SDIO pad control registers

* PAD\_CTRL\_1 Register (X\_SDIO0\_SDCLK, X\_SDIO0\_CMD, X\_SDIO0\_DAT \_0, X\_SDIO0\_DAT \_1)

Address: Operational Base + offset (0x04)



Figure 4. PAD\_CTRL\_1 Register

Table 4. PAD\_CTRL\_1 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_SDIO0\_SDCLK | 31 | RW | 0x0 | Function enable signal, high active |
| 30:28 | - | - | Reserved |
| 27 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 26 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 25:24 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO0\_CMD | 23 | RW | 0x0 | Function enable signal, high active |
| 22:20 | - | - | Reserved |
| 19 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17:16 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO0\_DAT \_0 | 15 | RW | 0x0 | Function enable signal, high active |
| 14:12 | - | - | Reserved |
| 11 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 10 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 9:8 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO0\_DAT \_1 | 7 | RW | 0x0 | Function enable signal, high active |
| 6:4 | - | - | Reserved |
| 3 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 2 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 1:0 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |

* PAD\_CTRL\_2 Register (X\_SDIO0\_DAT \_2, X\_SDIO0\_DAT \_3)

Address: Operational Base + offset (0x08)

Table 5. PAD\_CTRL\_2 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_SDIO0\_DAT \_2 | 31 | RW | 0x0 | Function enable signal, high active |
| 30:28 | - | - | Reserved |
| 27 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 26 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 25:24 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO0\_DAT \_3 | 23 | RW | 0x0 | Function enable signal, high active |
| 22:20 | - | - | Reserved |
| 19 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17:16 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| Reserved | 15:0 | - | - | Reserved |

* PAD\_CTRL\_3 Register (X\_SDIO1\_SDCLK , X\_SDIO1\_CMD, X\_SDIO1\_DAT \_0, X\_SDIO1\_DAT \_1)

Address: Operational Base + offset (0x0c)

Table 6. PAD\_CTRL\_3 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_SDIO1\_SDCLK | 31 | RW | 0x0 | Function enable signal, high active |
| 30:28 | - | - | Reserved |
| 27 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 26 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 25:24 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO1\_CMD | 23 | RW | 0x0 | Function enable signal, high active |
| 22:20 | - | - | Reserved |
| 19 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17:16 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO1\_DAT \_0 | 15 | RW | 0x0 | Function enable signal, high active |
| 14:12 | - | - | Reserved |
| 11 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 10 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 9:8 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO1\_DAT \_1 | 7 | RW | 0x0 | Function enable signal, high active |
| 6:4 | - | - | Reserved |
| 3 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 2 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 1:0 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |

* PAD\_CTRL\_4 Register (X\_SDIO1\_DAT \_2, X\_SDIO1\_DAT \_3)

Address: Operational Base + offset (0x10)

Table 7. PAD\_CTRL\_4 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_SDIO1\_DAT \_2 | 31 | RW | 0x0 | Function enable signal, high active |
| 30:28 | - | - | Reserved |
| 27 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 26 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 25:24 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| X\_SDIO1\_DAT \_3 | 23 | RW | 0x0 | Function enable signal, high active |
| 22:20 | - | - | Reserved |
| 19 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17:16 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| Reserved | 15:0 | - | - | Reserved |

**GMAC Pad Control Register**



Figure 5. Bit map for GMAC pad control registers

* PAD\_CTRL\_5 Register (X\_GMAC\_CLKT, X\_GMAC\_CLKR, X\_GMAC\_PHY\_50M, X\_GMAC\_PHY\_125M)

Address: Operational Base + offset (0x14)



Figure 6. PAD\_CTRL\_5 Register

Table 8. PAD\_CTRL\_5 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_CLKT | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_CLKR | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_PHY\_50M | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_PHY\_125M | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_6 Register (X\_GMAC\_CLKO, X\_GMAC\_TXD\_0, X\_GMAC\_TXD\_1, X\_GMAC\_TXD\_2)

Address: Operational Base + offset (0x18)

Table 9. PAD\_CTRL\_6 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_CLKO | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXD\_0 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXD\_1 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXD\_2 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_7 Register (X\_GMAC\_TXD\_3, X\_GMAC\_TXD\_4, X\_GMAC\_TXD\_5, X\_GMAC\_TXD\_6)

Address: Operational Base + offset (0x1c)

Table 10. PAD\_CTRL\_7 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_TXD\_3 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXD\_4 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXD\_5 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXD\_6 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_8 Register (X\_GMAC\_TXD\_7, X\_GMAC\_MDIO, X\_GMAC\_MDC, X\_GMAC\_RXER)

Address: Operational Base + offset (0x20)

Table 11. PAD\_CTRL\_8 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_TXD\_7 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_MDIO | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_MDC | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXER | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_9 Register (X\_GMAC\_RXDV, X\_GMAC\_COL, X\_GMAC\_CRS, X\_GMAC\_RXD\_0)

Address: Operational Base + offset (0x24)

Table 12. PAD\_CTRL\_9 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_RXDV | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_COL | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_CRS | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXD\_0 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_10 Register (X\_GMAC\_RXD\_1, X\_GMAC\_RXD\_2, X\_GMAC\_RXD\_3, X\_GMAC\_RXD\_4)

Address: Operational Base + offset (0x28)

Table 13. PAD\_CTRL\_10 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_RXD\_1 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXD\_2 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXD\_3 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXD\_4 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_11 Register (X\_GMAC\_RXD\_5, X\_GMAC\_RXD\_6, X\_GMAC\_RXD\_7, X\_GMAC\_TXEN)

Address: Operational Base + offset (0x2c)

Table 14. PAD\_CTRL\_11 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_GMAC\_RXD\_5 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | - | - | Reserved |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXD\_6 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | - | - | Reserved |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_RXD\_7 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | - | - | Reserved |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GMAC\_TXEN | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_12 Register (X\_GMAC\_TXER)

Address: Operational Base + offset (0x30)

Table 15. PAD\_CTRL\_12 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:8 | - | - | Reserved |
| X\_GMAC\_TXER | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | - | - | Reserved |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

**Dedicated GPIO Pad Control Register**



Figure 7. Bit map for dedicated GPIO pad control registers

* PAD\_CTRL\_13 Register (X\_SYSPWREN, X\_WAKE\_IRQ, X\_SYS\_RST\_N)

Address: Operational Base + offset (0x34)



Figure 8. PAD\_CTRL\_13 Register

Table 16. PAD\_CTRL\_13 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:24 | - | - | Reserved |
| X\_SYSPWREN | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_WAKE\_IRQ | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SYS\_RST\_N | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_14 Register (X\_RTC\_SCL, X\_RTC\_SDA, X\_RTC\_INT, X\_RTC\_32K\_CLK)

Address: Operational Base + offset (0x38)

Table 17. PAD\_CTRL\_14 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_RTC\_SCL | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_RTC\_SDA | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_RTC\_INT | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_RTC\_32K\_CLK | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_15 Register

Address: Operational Base + offset (0x3c)

Table 18. PAD\_CTRL\_15 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:0 | - | - | Reserved |

* PAD\_CTRL\_16 Register

Address: Operational Base + offset (0x40)

Table 19. PAD\_CTRL\_16 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:0 | - | - | Reserved |

* PAD\_CTRL\_17 Register (X\_IVA\_EXTCLK, X\_IVA\_PIXCLK, X\_IVA\_ FRAME\_VALID, X\_ LINE\_VALID)

Address: Operational Base + offset (0x44)

Table 20. PAD\_CTRL\_17 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_IVA\_EXTCLK | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_PIXCLK | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_ FRAME\_VALID | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_ LINE\_VALID | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_18 Register (X\_IVA\_DATA\_0, X\_IVA\_DATA\_1, X\_IVA\_DATA\_2, X\_DATA\_3)

Address: Operational Base + offset (0x48)

Table 21. PAD\_CTRL\_18 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_IVA\_DATA\_0 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_1 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_2 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_3 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_19 Register (X\_IVA\_DATA\_4, X\_IVA\_DATA\_5, X\_IVA\_DATA\_6, X\_DATA\_7)

Address: Operational Base + offset (0x4c)

Table 22. PAD\_CTRL\_19 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_IVA\_DATA\_4 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_5 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_6 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_7 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_20 Register (X\_IVA\_DATA\_8, X\_IVA\_DATA\_9, X\_IVA\_DATA\_10, X\_DATA\_11)

Address: Operational Base + offset (0x50)

Table 23. PAD\_CTRL\_20 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_IVA\_DATA\_8 | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_9 | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_10 | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_IVA\_DATA\_11 | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_21 Register (X\_SDIO0\_SDCD\_N, X\_SDIO0\_SDWP\_N)

Address: Operational Base + offset (0x54)

Table 24. PAD\_CTRL\_21 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:16 | - | - | Reserved |
| X\_SDIO0\_SDCD\_N | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO0\_SDWP\_N | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_22 Register (X\_SDIO0\_CLE, X\_SDIO0\_LED, X\_SDIO0\_BUS\_POW, X\_SDIO0\_BUS\_VOLT)

Address: Operational Base + offset (0x58)

Table 25. PAD\_CTRL\_22 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| X\_SDIO0\_CLE | 31 | RW | 0x0 | Function enable signal, high active |
| 30 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 29 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 28:27 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 26 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 25:24 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO0\_LED | 23 | RW | 0x0 | Function enable signal, high active |
| 22 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 21 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 20:19 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 18 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 17:16 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO0\_BUS\_POW | 15 | RW | 0x0 | Function enable signal, high active |
| 14 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 13 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 12:11 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 10 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 9:8 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO0\_BUS\_VOLT | 7 | RW | 0x0 | Function enable signal, high active |
| 6 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

**Alternated GPIO Pad Control Register**



Figure 9. Bit map for alternated GPIO pad control registers

* PAD\_CTRL\_23 Register (X\_SDIO1\_SDCD\_N, X\_SDIO1\_SDWP\_N, X\_SDIO1\_CLE)

Address: Operational Base + offset (0x5c)



Figure 10. PAD\_CTRL\_23 Register

Table 26. PAD\_CTRL\_23 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_SDIO1\_SDCD\_N | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO1\_SDWP\_N | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO1\_CLE | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_24 Register (X\_SDIO1\_LED, X\_SDIO1\_BUS\_POW, X\_SDIO1\_BUS\_VOLT)

Address: Operational Base + offset (0x60)

Table 27. PAD\_CTRL\_24 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_SDIO1\_LED | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO1\_BUS\_POW | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SDIO1\_BUS\_VOLT | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_25 Register (X\_PWM\_0, X\_PWM\_1, X\_PWM\_2)

Address: Operational Base + offset (0x64)

Table 28. PAD\_CTRL\_25 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_PWM\_0 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_PWM\_1 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_PWM\_2 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_26 Register (X\_SPI0\_SCLK, X\_SPI0\_MOSI, X\_SPI0\_MISO)

Address: Operational Base + offset (0x68)

Table 29. PAD\_CTRL\_26 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_SPI0\_SCLK | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI0\_MOSI | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI0\_MISO | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_27 Register (X\_SPI0\_SS\_N, X\_SPI1\_SCLK, X\_SPI1\_MOSI)

Address: Operational Base + offset (0x6c)

Table 30. PAD\_CTRL\_27 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_SPI0\_SS\_N | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI1\_SCLK | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI1\_MOSI | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_28 Register (X\_SPI1\_MISO, X\_SPI1\_SS\_N)

Address: Operational Base + offset (0x70)

Table 31. PAD\_CTRL\_28 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:20 | - | - | Reserved |
| X\_SPI1\_MISO | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI1\_SS\_N | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_29 Register (X\_I2C\_SCL, X\_I2C\_SDA)

Address: Operational Base + offset (0x74)

Table 32. PAD\_CTRL\_29 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:20 | - | - | Reserved |
| X\_I2C\_SCL | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_I2C\_SDA | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_30 Register (X\_UART\_TXD, X\_UART\_RXD)

Address: Operational Base + offset (0x78)

Table 33. PAD\_CTRL\_30 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:20 | - | - | Reserved |
| X\_UART\_TXD | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_UART\_RXD | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_31 Register (X\_GPIO\_PA\_0, X\_GPIO\_PA\_1, X\_GPIO\_PA\_2)

Address: Operational Base + offset (0x7c)

Table 34. PAD\_CTRL\_31 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_GPIO\_PA\_0 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PA\_1 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PA\_2 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_32 Register (X\_GPIO\_PA\_3, X\_GPIO\_PA\_4, X\_GPIO\_PA\_5)

Address: Operational Base + offset (0x80)

Table 35. PAD\_CTRL\_32 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_GPIO\_PA\_3 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PA\_4 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PA\_5 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_33 Register (X\_GPIO\_PA\_6, X\_GPIO\_PA\_7)

Address: Operational Base + offset (0x84)

Table 36. PAD\_CTRL\_33 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:20 | - | - | Reserved |
| X\_GPIO\_PA\_6 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PA\_7 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_34 Register (X\_GPIO\_PB\_0, X\_GPIO\_PB\_1, X\_GPIO\_PB\_2)

Address: Operational Base + offset (0x88)

Table 37. PAD\_CTRL\_34 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_GPIO\_PB\_0 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PB\_1 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PB\_2 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_35 Register (X\_GPIO\_PB\_3, X\_GPIO\_PB\_4, X\_GPIO\_PB\_5)

Address: Operational Base + offset (0x8c)

Table 38. PAD\_CTRL\_35 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_GPIO\_PB\_3 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PB\_4 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PB\_5 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_36 Register (X\_GPIO\_PB\_6, X\_GPIO\_PB\_7)

Address: Operational Base + offset (0x90)

Table 39. PAD\_CTRL\_36 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:20 | - | - | Reserved |
| X\_GPIO\_PB\_6 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PB\_7 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_35 Register (X\_GPIO\_PC\_0, X\_GPIO\_PC\_1, X\_GPIO\_PC\_2)

Address: Operational Base + offset (0x8c)

Table 40. PAD\_CTRL\_35 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_GPIO\_PC\_0 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PC\_1 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PC\_2 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_38 Register (X\_GPIO\_PC\_3, X\_GPIO\_PC\_4, X\_GPIO\_PC\_5)

Address: Operational Base + offset (0x98)

Table 41. PAD\_CTRL\_38 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_GPIO\_PC\_3 | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PC\_4 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PC\_5 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_39 Register (X\_GPIO\_PC\_6, X\_GPIO\_PC\_7)

Address: Operational Base + offset (0x9c)

Table 42. PAD\_CTRL\_39 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:20 | - | - | Reserved |
| X\_GPIO\_PC\_6 | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_GPIO\_PC\_7 | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_40 Register (X\_SPI\_NOR\_SCLK, X\_SPI\_NOR\_MOSI, X\_SPI\_NOR\_MISO)

Address: Operational Base + offset (0xa0)

Table 43. PAD\_CTRL\_40 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_SPI\_NOR\_SCLK | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI\_NOR\_MOSI | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI\_NOR\_MISO | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* PAD\_CTRL\_41 Register (X\_SPI\_NOR\_SS\_N, X\_SPI\_NOR\_SIO2\_IN, X\_SPI\_NOR\_SIO3\_IN)

Address: Operational Base + offset (0xa4)

Table 44. PAD\_CTRL\_41 Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAD Name | Bit | Attr | Reset Value | Description |
| Reserved | 31:30 | - | - | Reserved |
| X\_SPI\_NOR\_SS\_N | 29 | RW | 0x0 | Function enable signal, high active |
| 28 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 27 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 26 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0 (C: Data output to the core)  1 : receiver enable |
| 25 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 24:23 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 22 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 21:20 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI\_NOR\_SIO2\_IN | 19 | RW | 0x0 | Function enable signal, high active |
| 18 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 17 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 16 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 15 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 14:13 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 12 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 11:10 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |
| X\_SPI\_NOR\_SIO3\_IN | 9 | RW | 0x0 | Function enable signal, high active |
| 8 | RW | 0x0 | SRC : Slew Rate Control  0 : slow (half frequency)  1 : fast |
| 7 | RW | 0x0 | OEN : active low output driver enable  1 : driver disable  0 : normal operation |
| 6 | RW | 0x0 | REN : active high receiver enable,  0 : receiver disable, C driven to 0  1 : receiver enable |
| 5 | RW | 0x0 | SMT: active high Schmitt select  0 : no hysteresis  1 : Schmitt trigger |
| 4:3 | RW | 0x0 | Driver disabled state control  00:Hi-Z (Normal operation)  01:Pull-up  10:Pull-down  11:Repeater (Bus keeper) |
| 2 | RW | 0x0 | POS : Power-on-Start enable  1 : enable weak pull-down for invalid for invalid power  0 : weak pull down disable |
| 1:0 | RW | 0x0 | Driving Strength, 00:2mA, 01:4mA, 10:8mA, 11:12mA |

* + 1. Engineering Mode Select Register
* SW\_MUX\_SEL Register

Address: Operational Base + offset (0x100)

Table 45. SW\_MUX\_SEL Register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Attr | Reset Value | Description |
| 31:4 | - | - | Reserved |
| 3:0 | RW | 0x0 | SW MUX select signal:  4’b0000: CoreSight Trace Mode  4’b0001: DDR\_DFI Debug Mode  4’b0010: DDR\_PAD Debug Mode  4’b0011: SDIO Debug Mode  4’b0100: GMAC Debug Mode  4’b0101: ISRAM Debug Mode  4’b0110: DMA Debug Mode  4’b0111: CoreSight Debug Mode  4’b1000: SPI Debug Mode  4’b1001: UART Debug Mode  4’b1010: I2C Debug Mode  4’b1011: SPI-NOR Debug Mode  4’b1100: IVA Debug Mode  4’b1111: CoreSight JTAG Mode |

* DBG\_BANK\_SEL0 Register

Address: Operational Base + offset (0x104)

Table 46. DBG\_BANK\_SEL0 Register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Attr | Reset Value | Description |
| 31:29 | - | - | Reserved |
| 28 | RW | 0x0 | DBG\_IVA\_DBGMODE, switching IVA debug signals  0: IVA debug scale index signals  1: IVA cfm debug signals |
| 27:25 | - | - | Reserved |
| 24 | RW | 0x0 | reg\_i2c\_debug\_sel, selecting I2C (I2C0 or I2C1) debug signals  0: I2C0 debug signals(it contains data , oen slave address and master state)  1: I2C1 debug signals(it contains data , oen slave address and master state) |
| 23:22 | - | - | Reserved |
| 21 | RW | 0x0 | DBG\_SPI\_DBGMODE, switching the bank of SPI debug signals  0: Debug serial data and state machine  1: Debug tx fifo and rx fifo input data |
| 20 | RW | 0x0 | reg\_spi\_debug\_sel, selecting SPI (SPI0 or SPI1) debug signals  0: SPI0 debug signals  1: SPI1 debug signals |
| 19:17 | - | - | Reserved |
| 16 | RW | 0x0 | DBG\_ISRAM\_DBGMODE, switching ISRAM debug signals  0: ISRAM0 debug the interface 0 signals (it contains data、wen、cen and address)  1: ISRAM1 debug the interface 1 signals(it contains data、wen、cen and address) |
| 15:14 | - | - | Reserved |
| 13:12 | RW | 0x0 | DBG\_DMA\_DBGMODE, switching DMA debug signals  0: Debug the UART request interface (it contains tx & rx request signals).  1: Debug the SPI0 & SPI1 request interface (it contains tx & rx request signals)  2: Debug the SPINOR request interface (it contains tx & rx request signals) |
| 11:9 | - | - | Reserved |
| 8 | RW | 0x0 | DBG\_SDIO\_DBGMODE, switching SDIO debug signals  0: Debug SD data transmission without using DMA mechanism  1: Debug SD data transmission using DMA mechanism and clock tuning |
| 7:6 | - | - | Reserved |
| 5:4 | RW | 0x0 | DBG\_DDR\_DBGMODE\_B, switching DDR debug mode B signals  2’b00: PAD\_WRL mode ; Select the mode to mainly monitor Clock/address/data strobe/lower 16 bits write data before through IO PAD, From DDR PHY to device, and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*).  2’b01: PAD\_WRH mode; Select the mode to mainly monitor DDR PHY Clock/address/data strobe /higher 16 bits write data before through IO PAD From DDR PHY to device, and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*).  2’b10: PAD\_RDL mode; Select the mode to mainly monitor DDR PHY Clock/address/data strobe /lower 16 bits read data after through IO PAD From device to DDR PHY, and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*).  2’b11: PAD\_RDH mode; Select the mode to mainly monitor DDR PHY Clock/address/data strobe /lower 16 bits read data after through IO PAD from device to DDR PHY ,and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*). |
| 3:0 | RW | 0x0 | DBG\_DDR\_DBGMODE\_A, switching DDR debug mode B signals  4’b0000: DFI\_CA mode; Select the mode to switch to mainly monitor Part key command signals of DFI protocol.  4’b0001: DFI\_WR0 mode; Select the mode to mainly monitor two lowest (1st)byte of write data channel signals of DFI Protocol.  4’b0010: DFI\_WR1 mode; Select the mode to mainly monitor two (2st)byte of write data channel signals of DFI Protocol.  4’b0011: DFI\_WR2 mode; Select the mode to mainly monitor two lowest (3st)byte of write data channel signals of DFI Protocol.  4’b0100: DFI\_WR3 mode; Select the mode to mainly monitor two lowest (4st)byte of write data channel signals of DFI Protocol.  4’b0101: DFI\_RD0 mode; Select the mode to mainly monitor two lowest (1st)byte of Read data channel signals of DFI Protocol, and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*).  4’b0110: DFI\_RD1 mode; Select the mode to mainly monitor two lowest (2st)byte of Read data channel signals of DFI Protocol, and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*).  4’b0111: DFI\_RD2 mode; Select the mode to mainly monitor two lowest (3st)byte of Read data channel signals of DFI Protocol ,and DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*).  4’b1000: DFI\_RD3 mode; Select the mode to mainly monitor two lowest (4st)byte of Read data channel signals of DFI Protocol, DFI status signal(ex: *dfi\_ctrlupd\_req, dfi\_init\_complete*). |

* DBG\_BANK\_SEL1 Register

Address: Operational Base + offset (0x108)

Table 47. DBG\_BANK\_SEL1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Attr | Reset Value | Description |
| 31:2 | - | - | Reserved |
| 1:0 | RW | 0x0 | DBG\_UART\_DBGMODE, switching UART debug signals  2’b00: debug dma signal status and tx & rx fifo status  2’b01: debug tx & rx fifo state machine  2’b10: debug interrupt flag and txd & rxd serial data |

* + 1. PAD Voltage Control Register

Address: Operational Base + offset (0x110)

Table 48. PAD\_VOLT\_CTRL Register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Attr | Reset Value | Description |
| 31:8 | - | - | Reserved |
| 7 | RW | 0x0 | Function enable signal, high active |
| 6:5 | - | - | Reserved |
| 4 | RW | 0x0 | Video-in active high 1.8 V range select   * SEL18 = 0 – I/O domain voltage ≥ 2.5V * SEL18 = 1 – I/O domain voltage ≤ 1.8V |
| 3 | RW | 0x0 | Function enable signal, high active |
| 2:1 | - | - | Reserved |
| 0 | RW | 0x0 | GMAC active high 1.8 V range select   * SEL18 = 0 – I/O domain voltage ≥ 2.5V * SEL18 = 1 – I/O domain voltage ≤ 1.8V |

* + 1. CoreSight Trace Select Register

Address: Operational Base + offset (0x114)

Table 49. CS\_TRACE\_SEL Register

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Attr | Reset Value | Description |
| 31:1 | - | - | Reserved |
| 0 | RW | 0x0 | CoreSight trace signal delay enable  0: CoreSight trace signals output with delaying 1T (trace clock)  1: CoreSight trace signal output directly |

* 1. IOMUX Table
     1. Normal & Engineering & FPGA Mode

Table 50. Normal & Engineering & FPGA Mode IOMUX Table

Note: DDR has its own dedicated IO PADs, so DDR PADs are not included in IOMUX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO. | Common Pin Name | Normal Mode | Engineering Mode | | FPGA Test Mode | | |
| MODE\_SEL = 4’h0 | MODE\_SEL = 4’h1 &  reg\_mux\_sel = 4’h0 | MODE\_SEL = 4’h1 &  reg\_mux\_sel = 4’hf | MODE\_SEL = 4’hf &  reg\_mux\_sel = 4’h0 | MODE\_SEL = 4’hf &  reg\_mux\_sel = 4’h1 | MODE\_SEL = 4’hf &  reg\_mux\_sel = 4’hf |
| Normal Mode  IVA PAD power = 2.8V | CoreSight Trace Mode  IVA PAD power = 2.8V | CoreSight JTAG Mode  IVA PAD power = 2.8V | Normal Mode  IVA PAD power = 1.8V | CoreSight Trace Mode  IVA PAD power = 1.8V | CoreSight Trace Mode  IVA PAD power = 1.8V |
| 1 | X\_POR\_N | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I |
| 2 | X\_SYS\_RST\_N | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O |
| 3 | X\_SYSPWREN | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O |
| 4 | X\_WAKE\_IRQ | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I |
| 5 | X\_MAIN\_CLK\_I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I |
| 6 | X\_MAIN\_CLK\_O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O |
| 7 | X\_RTC\_SCL | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B |
| 8 | X\_RTC\_SDA | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B |
| 9 | X\_RTC\_INT | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I |
| 10 | X\_RTC\_32K\_CLK | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I |
| 11 | X\_MODE\_SEL\_0 | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I |
| 12 | X\_MODE\_SEL\_1 | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I |
| 13 | X\_MODE\_SEL\_2 | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I |
| 14 | X\_MODE\_SEL\_3 | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I |
| 15 | X\_BOOT\_0 | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I |
| 16 | X\_BOOT\_1 | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I |
| 17 | X\_DDR\_CLK | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O |
| 18 | X\_DDR\_CLK\_N | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O |
| 19 | X\_DDR\_DATA\_0 | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B |
| 20 | X\_DDR\_DATA\_1 | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B |
| 21 | X\_DDR\_DATA\_2 | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B |
| 22 | X\_DDR\_DATA\_3 | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B |
| 23 | X\_DDR\_DATA\_4 | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B |
| 24 | X\_DDR\_DATA\_5 | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B |
| 25 | X\_DDR\_DATA\_6 | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B |
| 26 | X\_DDR\_DATA\_7 | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B |
| 27 | X\_DDR\_DATA\_8 | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B |
| 28 | X\_DDR\_DATA\_9 | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B |
| 29 | X\_DDR\_DATA\_10 | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B |
| 30 | X\_DDR\_DATA\_11 | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B |
| 31 | X\_DDR\_DATA\_12 | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B |
| 32 | X\_DDR\_DATA\_13 | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B |
| 33 | X\_DDR\_DATA\_14 | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B |
| 34 | X\_DDR\_DATA\_15 | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B |
| 35 | X\_DDR\_DATA\_16 | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B |
| 36 | X\_DDR\_DATA\_17 | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B |
| 37 | X\_DDR\_DATA\_18 | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B |
| 38 | X\_DDR\_DATA\_19 | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B |
| 39 | X\_DDR\_DATA\_20 | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B |
| 40 | X\_DDR\_DATA\_21 | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B |
| 41 | X\_DDR\_DATA\_22 | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B |
| 42 | X\_DDR\_DATA\_23 | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B |
| 43 | X\_DDR\_DATA\_24 | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B |
| 44 | X\_DDR\_DATA\_25 | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B |
| 45 | X\_DDR\_DATA\_26 | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B |
| 46 | X\_DDR\_DATA\_27 | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B |
| 47 | X\_DDR\_DATA\_28 | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B |
| 48 | X\_DDR\_DATA\_29 | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B |
| 49 | X\_DDR\_DATA\_30 | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B |
| 50 | X\_DDR\_DATA\_31 | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B |
| 51 | X\_DDR\_ADDR\_0 | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O |
| 52 | X\_DDR\_ADDR\_1 | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O |
| 53 | X\_DDR\_ADDR\_2 | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O |
| 54 | X\_DDR\_ADDR\_3 | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O |
| 55 | X\_DDR\_ADDR\_4 | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O |
| 56 | X\_DDR\_ADDR\_5 | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O |
| 57 | X\_DDR\_ADDR\_6 | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O |
| 58 | X\_DDR\_ADDR\_7 | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O |
| 59 | X\_DDR\_ADDR\_8 | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O |
| 60 | X\_DDR\_ADDR\_9 | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O |
| 61 | X\_DDR \_CKE | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O |
| 62 | X\_DDR\_CS\_N | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O |
| 63 | X\_DDR\_DM\_0 | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B |
| 64 | X\_DDR\_DM\_1 | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B |
| 65 | X\_DDR\_DM\_2 | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B |
| 66 | X\_DDR\_DM\_3 | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B |
| 67 | X\_DDR\_DQS\_M\_0 | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B |
| 68 | X\_DDR\_DQS\_M\_1 | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B |
| 69 | X\_DDR\_DQS\_M\_2 | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B |
| 70 | X\_DDR\_DQS\_M\_3 | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B |
| 71 | X\_DDR\_DQS\_P\_0 | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B |
| 72 | X\_DDR\_DQS\_P\_1 | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B |
| 73 | X\_DDR\_DQS\_P\_2 | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B |
| 74 | X\_DDR\_DQS\_P\_3 | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B |
| 75 | X\_DDR\_ODT | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O |
| 76 | X\_SDIO0\_SDCLK | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O |
| 77 | X\_SDIO0\_CMD | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B |
| 78 | X\_SDIO0\_DAT \_0 | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B |
| 79 | X\_SDIO0\_DAT \_1 | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B |
| 80 | X\_SDIO0\_DAT\_2 | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B |
| 81 | X\_SDIO0\_DAT \_3 | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B |
| 82 | X\_SDIO0\_SDCD\_N | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I |
| 83 | X\_SDIO0\_SDWP\_N | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I |
| 84 | X\_SDIO0\_CLE | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I |
| 85 | X\_SDIO0\_LED | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O |
| 86 | X\_SDIO0\_BUS\_POW | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O |
| 87 | X\_SDIO0\_BUS\_VOLT | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O |
| 88 | X\_SDIO1\_SDCLK | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O |
| 89 | X\_SDIO1\_CMD | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B |
| 90 | X\_SDIO1\_DAT\_0 | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B |
| 91 | X\_SDIO1\_DAT\_1 | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B |
| 92 | X\_SDIO1\_DAT\_2 | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B |
| 93 | X\_SDIO1\_DAT\_3 | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B |
| 94 | X\_SDIO1\_SDCD\_N | SDIO1\_SDCD\_N /I | SDIO1\_SDCD\_N /I | SDIO1\_SDCD\_N /I | SDIO1\_SDCD\_N /I | SDIO1\_SDCD\_N /I | SDIO1\_SDCD\_N /I |
| 95 | X\_SDIO1\_SDWP\_N | SDIO1\_SDWP\_N /I | SDIO1\_SDWP\_N /I | SDIO1\_SDWP\_N /I | SDIO1\_SDWP\_N /I | SDIO1\_SDWP\_N /I | SDIO1\_SDWP\_N /I |
| 96 | X\_SDIO1\_CLE | SDIO1\_CLE/I | SDIO1\_CLE/I | SDIO1\_CLE/I | SDIO1\_CLE/I | SDIO1\_CLE/I | SDIO1\_CLE/I |
| 97 | X\_SDIO1\_LED | SDIO1\_LED/O | SDIO1\_LED/O | SDIO1\_LED/O | SDIO1\_LED/O | SDIO1\_LED/O | SDIO1\_LED/O |
| 98 | X\_SDIO1\_BUS\_POW | SDIO1\_BUS\_POW/O | SDIO1\_BUS\_POW/O | SDIO1\_BUS\_POW/O | SDIO1\_BUS\_POW/O | SDIO1\_BUS\_POW/O | SDIO1\_BUS\_POW/O |
| 99 | X\_SDIO1\_BUS\_VOLT | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O |
| 100 | X\_GMAC\_CLKT | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I |
| 101 | X\_GMAC\_CLKR | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I |
| 102 | X\_GMAC\_PHY\_50M | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I |
| 103 | X\_GMAC\_PHY\_125M | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I |
| 104 | X\_GMAC\_CLKO | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O |
| 105 | X\_GMAC\_TXD\_0 | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O |
| 106 | X\_GMAC\_TXD\_1 | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O |
| 107 | X\_GMAC\_TXD\_2 | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O |
| 108 | X\_GMAC\_TXD\_3 | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O |
| 109 | X\_GMAC\_TXD\_4 | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O |
| 110 | X\_GMAC\_TXD\_5 | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O |
| 111 | X\_GMAC\_TXD\_6 | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O |
| 112 | X\_GMAC\_TXD\_7 | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O |
| 113 | X\_GMAC\_MDIO | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B |
| 114 | X\_GMAC\_MDC | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O |
| 115 | X\_GMAC\_RXER | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I |
| 116 | X\_GMAC\_RXDV | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I |
| 117 | X\_GMAC\_COL | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I |
| 118 | X\_GMAC\_CRS | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I |
| 119 | X\_GMAC\_RXD\_0 | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I |
| 120 | X\_GMAC\_RXD\_1 | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I |
| 121 | X\_GMAC\_RXD\_2 | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I |
| 122 | X\_GMAC\_RXD\_3 | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I |
| 123 | X\_GMAC\_RXD\_4 | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I |
| 124 | X\_GMAC\_RXD\_5 | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I |
| 125 | X\_GMAC\_RXD\_6 | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I |
| 126 | X\_GMAC\_RXD\_7 | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I |
| 127 | X\_GMAC\_TXEN | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O |
| 128 | X\_GMAC\_TXER | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O |
| 129 | X\_IVA\_EXTCLK | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O |
| 130 | X\_IVA\_PIXCLK | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I |
| 131 | X\_IVA\_FRAME\_VALID | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I |
| 132 | X\_IVA\_LINE\_VALID | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I |
| 133 | X\_IVA\_DATA\_0 | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I |
| 134 | X\_IVA\_DATA\_1 | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I |
| 135 | X\_IVA\_DATA\_2 | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I |
| 136 | X\_IVA\_DATA\_3 | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I |
| 137 | X\_IVA\_DATA\_4 | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I |
| 138 | X\_IVA\_DATA\_5 | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I |
| 139 | X\_IVA\_DATA\_6 | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I |
| 140 | X\_IVA\_DATA\_7 | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I |
| 141 | X\_IVA\_DATA\_8 | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I |
| 142 | X\_IVA\_DATA\_9 | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I |
| 143 | X\_IVA\_DATA\_10 | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I |
| 144 | X\_IVA\_DATA\_11 | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I |
| 145 | X\_PWM\_0 | PWM\_0/B | PWM\_0/B | PWM\_0/B | PWM\_0/B | PWM\_0/B | PWM\_0/B |
| 146 | X\_PWM\_1 | PWM\_1/B | CA7\_EDBGRQ/I | PWM\_1/B | PWM\_1/B | CA7\_EDBGRQ/I | PWM\_1/B |
| 147 | X\_PWM\_2 | PWM\_2/B | CA7\_DBGACK/O | PWM\_2/B | PWM\_2/B | CA7\_DBGACK/O | PWM\_2/B |
| 148 | X\_SPI0\_SCLK | SPI0\_SCLK/B | SPI0\_SCLK/B | SPI0\_SCLK/B | SPI0\_SCLK/B | SPI0\_SCLK/B | SPI0\_SCLK/B |
| 149 | X\_SPI0\_MOSI | SPI0\_MOSI/B | SPI0\_MOSI/B | SPI0\_MOSI/B | SPI0\_MOSI/B | SPI0\_MOSI/B | SPI0\_MOSI/B |
| 150 | X\_SPI0\_MISO | SPI0\_MISO/B | SPI0\_MISO/B | SPI0\_MISO/B | SPI0\_MISO/B | SPI0\_MISO/B | SPI0\_MISO/B |
| 151 | X\_SPI0\_SS\_N | SPI0\_SS\_N/B | SPI0\_SS\_N/B | SPI0\_SS\_N/B | SPI0\_SS\_N/B | SPI0\_SS\_N/B | SPI0\_SS\_N/B |
| 152 | X\_SPI1\_SCLK | SPI1\_SCLK/B | SPI1\_SCLK/B | SPI1\_SCLK/B | SPI1\_SCLK/B | SPI1\_SCLK/B | SPI1\_SCLK/B |
| 153 | X\_SPI1\_MOSI | SPI1\_MOSI/B | SPI1\_MOSI/B | SPI1\_MOSI/B | SPI1\_MOSI/B | SPI1\_MOSI/B | SPI1\_MOSI/B |
| 154 | X\_SPI1\_MISO | SPI1\_MISO/B | SPI1\_MISO/B | SPI1\_MISO/B | SPI1\_MISO/B | SPI1\_MISO/B | SPI1\_MISO/B |
| 155 | X\_SPI1\_SS\_N | SPI1\_SS\_N/B | SPI1\_SS\_N/B | SPI1\_SS\_N/B | SPI1\_SS\_N/B | SPI1\_SS\_N/B | SPI1\_SS\_N/B |
| 156 | X\_I2C\_SCL | I2C\_SCL/B | I2C\_SCL/B | I2C\_SCL/B | I2C\_SCL/B | I2C\_SCL/B | I2C\_SCL/B |
| 157 | X\_I2C\_SDA | I2C\_SDA/B | I2C\_SDA/B | I2C\_SDA/B | I2C\_SDA/B | I2C\_SDA/B | I2C\_SDA/B |
| 158 | X\_UART\_TXD | UART\_TXD/O | UART\_TXD/O | UART\_TXD/O | UART\_TXD/O | UART\_TXD/O | UART\_TXD/O |
| 159 | X\_UART\_RXD | UART\_RXD/I | UART\_RXD/I | UART\_RXD/I | UART\_RXD/I | UART\_RXD/I | UART\_RXD/I |
| 160 | X\_GPIO\_PA\_0 | GPIO\_PA\_0/B | tracedata\_trace\_out\_0/O | GPIO\_PA\_0/B | GPIO\_PA\_0/B | tracedata\_trace\_out\_0/O | GPIO\_PA\_0/B |
| 161 | X\_GPIO\_PA\_1 | GPIO\_PA\_1/B | tracedata\_trace\_out\_1/O | GPIO\_PA\_1/B | GPIO\_PA\_1/B | tracedata\_trace\_out\_1/O | GPIO\_PA\_1/B |
| 162 | X\_GPIO\_PA\_2 | GPIO\_PA\_2/B | tracedata\_trace\_out\_2/O | GPIO\_PA\_2/B | GPIO\_PA\_2/B | tracedata\_trace\_out\_2/O | GPIO\_PA\_2/B |
| 163 | X\_GPIO\_PA\_3 | GPIO\_PA\_3/B | tracedata\_trace\_out\_3/O | GPIO\_PA\_3/B | GPIO\_PA\_3/B | tracedata\_trace\_out\_3/O | GPIO\_PA\_3/B |
| 164 | X\_GPIO\_PA\_4 | GPIO\_PA\_4/B | tracedata\_trace\_out\_4/O | GPIO\_PA\_4/B | GPIO\_PA\_4/B | tracedata\_trace\_out\_4/O | GPIO\_PA\_4/B |
| 165 | X\_GPIO\_PA\_5 | GPIO\_PA\_5/B | tracedata\_trace\_out\_5/O | GPIO\_PA\_5/B | GPIO\_PA\_5/B | tracedata\_trace\_out\_5/O | GPIO\_PA\_5/B |
| 166 | X\_GPIO\_PA\_6 | GPIO\_PA\_6/B | tracedata\_trace\_out\_6/O | GPIO\_PA\_6/B | GPIO\_PA\_6/B | tracedata\_trace\_out\_6/O | GPIO\_PA\_6/B |
| 167 | X\_GPIO\_PA\_7 | GPIO\_PA\_7/B | tracedata\_trace\_out\_7/O | GPIO\_PA\_7/B | GPIO\_PA\_7/B | tracedata\_trace\_out\_7/O | GPIO\_PA\_7/B |
| 168 | X\_GPIO\_PB\_0 | GPIO\_PB\_0/B | tracedata\_trace\_out\_8/O | GPIO\_PB\_0/B | GPIO\_PB\_0/B | tracedata\_trace\_out\_8/O | GPIO\_PB\_0/B |
| 169 | X\_GPIO\_PB\_1 | GPIO\_PB\_1/B | tracedata\_trace\_out\_9/O | GPIO\_PB\_1/B | GPIO\_PB\_1/B | tracedata\_trace\_out\_9/O | GPIO\_PB\_1/B |
| 170 | X\_GPIO\_PB\_2 | GPIO\_PB\_2/B | tracedata\_trace\_out\_10/O | GPIO\_PB\_2/B | GPIO\_PB\_2/B | tracedata\_trace\_out\_10/O | GPIO\_PB\_2/B |
| 171 | X\_GPIO\_PB\_3 | GPIO\_PB\_3/B | tracedata\_trace\_out\_11/O | GPIO\_PB\_3/B | GPIO\_PB\_3/B | tracedata\_trace\_out\_11/O | GPIO\_PB\_3/B |
| 172 | X\_GPIO\_PB\_4 | GPIO\_PB\_4/B | tracedata\_trace\_out\_12/O | GPIO\_PB\_4/B | GPIO\_PB\_4/B | tracedata\_trace\_out\_12/O | GPIO\_PB\_4/B |
| 173 | X\_GPIO\_PB\_5 | GPIO\_PB\_5/B | tracedata\_trace\_out\_13/O | GPIO\_PB\_5/B | GPIO\_PB\_5/B | tracedata\_trace\_out\_13/O | GPIO\_PB\_5/B |
| 174 | X\_GPIO\_PB\_6 | GPIO\_PB\_6/B | tracedata\_trace\_out\_14/O | GPIO\_PB\_6/B | GPIO\_PB\_6/B | tracedata\_trace\_out\_14/O | GPIO\_PB\_6/B |
| 175 | X\_GPIO\_PB\_7 | GPIO\_PB\_7/B | tracedata\_trace\_out\_15/O | GPIO\_PB\_7/B | GPIO\_PB\_7/B | tracedata\_trace\_out\_15/O | GPIO\_PB\_7/B |
| 176 | X\_GPIO\_PC\_0 | GPIO\_PC\_0/B | traceclk\_trace\_out/O | CA7\_EDBGRQ/I | GPIO\_PC\_0/B | traceclk\_trace\_out/O | CA7\_EDBGRQ/I |
| 177 | X\_GPIO\_PC\_1 | GPIO\_PC\_1/B | tracectrl\_trace\_out/O | CA7\_DBGACK/O | GPIO\_PC\_1/B | tracectrl\_trace\_out/O | CA7\_DBGACK/O |
| 178 | X\_GPIO\_PC\_2 | GPIO\_PC\_2/B | CS\_ntrst/I | CS\_ntrst/I | GPIO\_PC\_2/B | CS\_ntrst/I | CS\_ntrst/I |
| 179 | X\_GPIO\_PC\_3 | GPIO\_PC\_3/B | CS\_swclktck/I | CS\_swclktck/I | GPIO\_PC\_3/B | CS\_swclktck/I | CS\_swclktck/I |
| 180 | X\_GPIO\_PC\_4 | GPIO\_PC\_4/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | GPIO\_PC\_4/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B |
| 181 | X\_GPIO\_PC\_5 | GPIO\_PC\_5/B | CS\_tdi\_swj/I | CS\_tdi\_swj/I | GPIO\_PC\_5/B | CS\_tdi\_swj/I | CS\_tdi\_swj/I |
| 182 | X\_GPIO\_PC\_6 | GPIO\_PC\_6/B | CS\_tdo\_swj/O | CS\_tdo\_swj/O | GPIO\_PC\_6/B | CS\_tdo\_swj/O | CS\_tdo\_swj/O |
| 183 | X\_GPIO\_PC\_7 | GPIO\_PC\_7/B | PD\_CPU\_DWN\_ACK/O | PD\_CPU\_DWN\_ACK/O | GPIO\_PC\_7/B | PD\_CPU\_DWN\_ACK/O | PD\_CPU\_DWN\_ACK/O |
| 184 | X\_SPI\_NOR\_SCLK | SPI\_NOR\_SCLK/B | SPI\_NOR\_SCLK/B | SPI\_NOR\_SCLK/B | SPI\_NOR\_SCLK/B | SPI\_NOR\_SCLK/B | SPI\_NOR\_SCLK/B |
| 185 | X\_SPI\_NOR\_MOSI | SPI\_NOR\_MOSI/B | SPI\_NOR\_MOSI/B | SPI\_NOR\_MOSI/B | SPI\_NOR\_MOSI/B | SPI\_NOR\_MOSI/B | SPI\_NOR\_MOSI/B |
| 186 | X\_SPI\_NOR\_MISO | SPI\_NOR\_MISO/B | SPI\_NOR\_MISO/B | SPI\_NOR\_MISO/B | SPI\_NOR\_MISO/B | SPI\_NOR\_MISO/B | SPI\_NOR\_MISO/B |
| 187 | X\_SPI\_NOR\_SS\_N | SPI\_NOR\_SS\_N/B | SPI\_NOR\_SS\_N/B | SPI\_NOR\_SS\_N/B | SPI\_NOR\_SS\_N/B | SPI\_NOR\_SS\_N/B | SPI\_NOR\_SS\_N/B |
| 188 | X\_SPI\_NOR\_SIO2\_IN | SPI\_NOR\_SIO2\_IN/I | SPI\_NOR\_SIO2\_IN/I | SPI\_NOR\_SIO2\_IN/I | SPI\_NOR\_SIO2\_IN/I | SPI\_NOR\_SIO2\_IN/I | SPI\_NOR\_SIO2\_IN/I |
| 189 | X\_SPI\_NOR\_SIO3\_IN | SPI\_NOR\_SIO3\_IN/I | SPI\_NOR\_SIO3\_IN/I | SPI\_NOR\_SIO3\_IN/I | SPI\_NOR\_SIO3\_IN/I | SPI\_NOR\_SIO3\_IN/I | SPI\_NOR\_SIO3\_IN/I |

* + 1. DFT & MBIST Mode

Table 51. DFT & MBIST Mode IOMUX Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Common Pin Name | TOP-SCAN MODE  (MODE\_SEL = 4’h2) | CPU-SCAN MODE  (MODE\_SEL = 4’h3) | TOP-MBIST MODE  (MODE\_SEL = 4’h4) | CPU-MBIST MODE  (MODE\_SEL = 4’h5) |
| 1 | X\_POR\_N | Scan\_rst/I | Reset /I | BIST Reset/I | BIST Reset/I |
| 2 | X\_SYS\_RST\_N | N.A. | N.A. | N.A. | N.A. |
| 3 | X\_SYSPWREN | N.A. | N.A. | N.A. | N.A. |
| 4 | X\_WAKE\_IRQ | N.A. | N.A. | N.A. | N.A. |
| 5 | X\_MAIN\_CLK\_I | Scan\_PLL\_clk/I | N.A. | N.A. | N.A. |
| 6 | X\_MAIN\_CLK\_O | N.A. | N.A. | N.A. | N.A. |
| 7 | X\_RTC\_SCL | N.A. | N.A. | N.A. | N.A. |
| 8 | X\_RTC\_SDA | N.A. | N.A. | N.A. | N.A. |
| 9 | X\_RTC\_INT | N.A. | N.A. | N.A. | N.A. |
| 10 | X\_RTC\_32K\_CLK | N.A. | N.A. | N.A. | N.A. |
| 11 | X\_MODE\_SEL\_0 | N.A. | N.A. | N.A. | N.A. |
| 12 | X\_MODE\_SEL\_1 | N.A. | N.A. | N.A. | N.A. |
| 13 | X\_MODE\_SEL\_2 | N.A. | N.A. | N.A. | N.A. |
| 14 | X\_MODE\_SEL\_3 | N.A. | N.A. | N.A. | N.A. |
| 15 | X\_BOOT\_0 | N.A. | N.A. | N.A. | N.A. |
| 16 | X\_BOOT\_1 | N.A. | N.A. | N.A. | N.A. |
| 17 | X\_DDR\_CLK | N.A. | N.A. | N.A. | N.A. |
| 18 | X\_DDR\_CLK\_N | N.A. | N.A. | N.A. | N.A. |
| 19 | X\_DDR\_DATA\_0 | N.A. | N.A. | N.A. | N.A. |
| 20 | X\_DDR\_DATA\_1 | N.A. | N.A. | N.A. | N.A. |
| 21 | X\_DDR\_DATA\_2 | N.A. | N.A. | N.A. | N.A. |
| 22 | X\_DDR\_DATA\_3 | N.A. | N.A. | N.A. | N.A. |
| 23 | X\_DDR\_DATA\_4 | N.A. | N.A. | N.A. | N.A. |
| 24 | X\_DDR\_DATA\_5 | N.A. | N.A. | N.A. | N.A. |
| 25 | X\_DDR\_DATA\_6 | N.A. | N.A. | N.A. | N.A. |
| 26 | X\_DDR\_DATA\_7 | N.A. | N.A. | N.A. | N.A. |
| 27 | X\_DDR\_DATA\_8 | N.A. | N.A. | N.A. | N.A. |
| 28 | X\_DDR\_DATA\_9 | N.A. | N.A. | N.A. | N.A. |
| 29 | X\_DDR\_DATA\_10 | N.A. | N.A. | N.A. | N.A. |
| 30 | X\_DDR\_DATA\_11 | N.A. | N.A. | N.A. | N.A. |
| 31 | X\_DDR\_DATA\_12 | N.A. | N.A. | N.A. | N.A. |
| 32 | X\_DDR\_DATA\_13 | N.A. | N.A. | N.A. | N.A. |
| 33 | X\_DDR\_DATA\_14 | N.A. | N.A. | N.A. | N.A. |
| 34 | X\_DDR\_DATA\_15 | N.A. | N.A. | N.A. | N.A. |
| 35 | X\_DDR\_DATA\_16 | N.A. | N.A. | N.A. | N.A. |
| 36 | X\_DDR\_DATA\_17 | N.A. | N.A. | N.A. | N.A. |
| 37 | X\_DDR\_DATA\_18 | N.A. | N.A. | N.A. | N.A. |
| 38 | X\_DDR\_DATA\_19 | N.A. | N.A. | N.A. | N.A. |
| 39 | X\_DDR\_DATA\_20 | N.A. | N.A. | N.A. | N.A. |
| 40 | X\_DDR\_DATA\_21 | N.A. | N.A. | N.A. | N.A. |
| 41 | X\_DDR\_DATA\_22 | N.A. | N.A. | N.A. | N.A. |
| 42 | X\_DDR\_DATA\_23 | N.A. | N.A. | N.A. | N.A. |
| 43 | X\_DDR\_DATA\_24 | N.A. | N.A. | N.A. | N.A. |
| 44 | X\_DDR\_DATA\_25 | N.A. | N.A. | N.A. | N.A. |
| 45 | X\_DDR\_DATA\_26 | N.A. | N.A. | N.A. | N.A. |
| 46 | X\_DDR\_DATA\_27 | N.A. | N.A. | N.A. | N.A. |
| 47 | X\_DDR\_DATA\_28 | N.A. | N.A. | N.A. | N.A. |
| 48 | X\_DDR\_DATA\_29 | N.A. | N.A. | N.A. | N.A. |
| 49 | X\_DDR\_DATA\_30 | N.A. | N.A. | N.A. | N.A. |
| 50 | X\_DDR\_DATA\_31 | N.A. | N.A. | N.A. | N.A. |
| 51 | X\_DDR\_ADDR\_0 | N.A. | N.A. | N.A. | N.A. |
| 52 | X\_DDR\_ADDR\_1 | N.A. | N.A. | N.A. | N.A. |
| 53 | X\_DDR\_ADDR\_2 | N.A. | N.A. | N.A. | N.A. |
| 54 | X\_DDR\_ADDR\_3 | N.A. | N.A. | N.A. | N.A. |
| 55 | X\_DDR\_ADDR\_4 | N.A. | N.A. | N.A. | N.A. |
| 56 | X\_DDR\_ADDR\_5 | N.A. | N.A. | N.A. | N.A. |
| 57 | X\_DDR\_ADDR\_6 | N.A. | N.A. | N.A. | N.A. |
| 58 | X\_DDR\_ADDR\_7 | N.A. | N.A. | N.A. | N.A. |
| 59 | X\_DDR\_ADDR\_8 | N.A. | N.A. | N.A. | N.A. |
| 60 | X\_DDR\_ADDR\_9 | N.A. | N.A. | N.A. | N.A. |
| 61 | X\_DDR \_CKE | N.A. | N.A. | N.A. | N.A. |
| 62 | X\_DDR\_CS\_N | N.A. | N.A. | N.A. | N.A. |
| 63 | X\_DDR\_DM\_0 | N.A. | N.A. | N.A. | N.A. |
| 64 | X\_DDR\_DM\_1 | N.A. | N.A. | N.A. | N.A. |
| 65 | X\_DDR\_DM\_2 | N.A. | N.A. | N.A. | N.A. |
| 66 | X\_DDR\_DM\_3 | N.A. | N.A. | N.A. | N.A. |
| 67 | X\_DDR\_DQS\_M\_0 | N.A. | N.A. | N.A. | N.A. |
| 68 | X\_DDR\_DQS\_M\_1 | N.A. | N.A. | N.A. | N.A. |
| 69 | X\_DDR\_DQS\_M\_2 | N.A. | N.A. | N.A. | N.A. |
| 70 | X\_DDR\_DQS\_M\_3 | N.A. | N.A. | N.A. | N.A. |
| 71 | X\_DDR\_DQS\_P\_0 | N.A. | N.A. | N.A. | N.A. |
| 72 | X\_DDR\_DQS\_P\_1 | N.A. | N.A. | N.A. | N.A. |
| 73 | X\_DDR\_DQS\_P\_2 | N.A. | N.A. | N.A. | N.A. |
| 74 | X\_DDR\_DQS\_P\_3 | N.A. | N.A. | N.A. | N.A. |
| 75 | X\_DDR\_ODT | N.A. | N.A. | N.A. | N.A. |
| 76 | X\_SDIO0\_SDCLK | N.A. | N.A. | N.A. | N.A. |
| 77 | X\_SDIO0\_CMD | N.A. | N.A. | N.A. | N.A. |
| 78 | X\_SDIO0\_DAT \_0 | N.A. | N.A. | N.A. | N.A. |
| 79 | X\_SDIO0\_DAT \_1 | N.A. | N.A. | N.A. | N.A. |
| 80 | X\_SDIO0\_DAT\_2 | N.A. | N.A. | N.A. | N.A. |
| 81 | X\_SDIO0\_DAT \_3 | N.A. | N.A. | N.A. | N.A. |
| 82 | X\_SDIO0\_SDCD\_N | N.A. | N.A. | N.A. | N.A. |
| 83 | X\_SDIO0\_SDWP\_N | N.A. | N.A. | N.A. | N.A. |
| 84 | X\_SDIO0\_CLE | N.A. | N.A. | N.A. | N.A. |
| 85 | X\_SDIO0\_LED | N.A. | N.A. | N.A. | N.A. |
| 86 | X\_SDIO0\_BUS\_POW | N.A. | N.A. | N.A. | N.A. |
| 87 | X\_SDIO0\_BUS\_VOLT | N.A. | N.A. | N.A. | N.A. |
| 88 | X\_SDIO1\_SDCLK | N.A. | N.A. | N.A. | N.A. |
| 89 | X\_SDIO1\_CMD | N.A. | N.A. | N.A. | N.A. |
| 90 | X\_SDIO1\_DAT\_0 | N.A. | N.A. | N.A. | N.A. |
| 91 | X\_SDIO1\_DAT\_1 | N.A. | N.A. | N.A. | N.A. |
| 92 | X\_SDIO1\_DAT\_2 | N.A. | N.A. | N.A. | N.A. |
| 93 | X\_SDIO1\_DAT\_3 | N.A. | N.A. | N.A. | N.A. |
| 94 | X\_SDIO1\_SDCD\_N | N.A. | N.A. | N.A. | N.A. |
| 95 | X\_SDIO1\_SDWP\_N | N.A. | N.A. | N.A. | N.A. |
| 96 | X\_SDIO1\_CLE | N.A. | N.A. | N.A. | N.A. |
| 97 | X\_SDIO1\_LED | N.A. | N.A. | N.A. | N.A. |
| 98 | X\_SDIO1\_BUS\_POW | N.A. | N.A. | N.A. | N.A. |
| 99 | X\_SDIO1\_BUS\_VOLT | N.A. | N.A. | N.A. | N.A. |
| 100 | X\_GMAC\_CLKT | N.A. | N.A. | N.A. | N.A. |
| 101 | X\_GMAC\_CLKR | N.A. | N.A. | N.A. | N.A. |
| 102 | X\_GMAC\_PHY\_50M | N.A. | N.A. | N.A. | N.A. |
| 103 | X\_GMAC\_PHY\_125M | N.A. | N.A. | N.A. | N.A. |
| 104 | X\_GMAC\_CLKO | N.A. | N.A. | N.A. | N.A. |
| 105 | X\_GMAC\_TXD\_0 | N.A. | N.A. | N.A. | N.A. |
| 106 | X\_GMAC\_TXD\_1 | N.A. | N.A. | N.A. | N.A. |
| 107 | X\_GMAC\_TXD\_2 | N.A. | N.A. | N.A. | N.A. |
| 108 | X\_GMAC\_TXD\_3 | N.A. | N.A. | N.A. | N.A. |
| 109 | X\_GMAC\_TXD\_4 | N.A. | N.A. | N.A. | N.A. |
| 110 | X\_GMAC\_TXD\_5 | N.A. | N.A. | N.A. | N.A. |
| 111 | X\_GMAC\_TXD\_6 | N.A. | N.A. | N.A. | N.A. |
| 112 | X\_GMAC\_TXD\_7 | N.A. | N.A. | N.A. | N.A. |
| 113 | X\_GMAC\_MDIO | N.A. | N.A. | N.A. | N.A. |
| 114 | X\_GMAC\_MDC | N.A. | N.A. | N.A. | N.A. |
| 115 | X\_GMAC\_RXER | N.A. | N.A. | N.A. | N.A. |
| 116 | X\_GMAC\_RXDV | N.A. | N.A. | N.A. | N.A. |
| 117 | X\_GMAC\_COL | N.A. | N.A. | N.A. | N.A. |
| 118 | X\_GMAC\_CRS | N.A. | N.A. | N.A. | N.A. |
| 119 | X\_GMAC\_RXD\_0 | N.A. | N.A. | N.A. | N.A. |
| 120 | X\_GMAC\_RXD\_1 | N.A. | N.A. | N.A. | N.A. |
| 121 | X\_GMAC\_RXD\_2 | N.A. | N.A. | N.A. | N.A. |
| 122 | X\_GMAC\_RXD\_3 | N.A. | N.A. | N.A. | N.A. |
| 123 | X\_GMAC\_RXD\_4 | N.A. | N.A. | N.A. | N.A. |
| 124 | X\_GMAC\_RXD\_5 | N.A. | N.A. | N.A. | N.A. |
| 125 | X\_GMAC\_RXD\_6 | N.A. | N.A. | N.A. | N.A. |
| 126 | X\_GMAC\_RXD\_7 | N.A. | N.A. | N.A. | N.A. |
| 127 | X\_GMAC\_TXEN | N.A. | N.A. | N.A. | N.A. |
| 128 | X\_GMAC\_TXER | N.A. | N.A. | N.A. | N.A. |
| 129 | X\_IVA\_EXTCLK | N.A. | N.A. | N.A. | N.A. |
| 130 | X\_IVA\_PIXCLK | N.A. | N.A. | N.A. | N.A. |
| 131 | X\_IVA\_FRAME\_VALID | N.A. | N.A. | N.A. | N.A. |
| 132 | X\_IVA\_LINE\_VALID | N.A. | N.A. | N.A. | N.A. |
| 133 | X\_IVA\_DATA\_0 | N.A. | N.A. | N.A. | N.A. |
| 134 | X\_IVA\_DATA\_1 | N.A. | N.A. | N.A. | N.A. |
| 135 | X\_IVA\_DATA\_2 | N.A. | N.A. | N.A. | N.A. |
| 136 | X\_IVA\_DATA\_3 | N.A. | N.A. | N.A. | N.A. |
| 137 | X\_IVA\_DATA\_4 | N.A. | N.A. | N.A. | N.A. |
| 138 | X\_IVA\_DATA\_5 | N.A. | N.A. | N.A. | N.A. |
| 139 | X\_IVA\_DATA\_6 | N.A. | N.A. | N.A. | N.A. |
| 140 | X\_IVA\_DATA\_7 | N.A. | N.A. | N.A. | N.A. |
| 141 | X\_IVA\_DATA\_8 | N.A. | N.A. | N.A. | N.A. |
| 142 | X\_IVA\_DATA\_9 | N.A. | N.A. | N.A. | N.A. |
| 143 | X\_IVA\_DATA\_10 | N.A. | N.A. | N.A. | N.A. |
| 144 | X\_IVA\_DATA\_11 | N.A. | N.A. | N.A. | N.A. |
| 145 | X\_PWM\_0 | OCC\_bypass/I | OCC\_bypass/I | N.A. | OCC\_bypass/I |
| 146 | X\_PWM\_1 | AtSpeed\_mode/I | AtSpeed\_mode/I | AtSpeed\_mode/I | AtSpeed\_mode/I |
| 147 | X\_PWM\_2 | scan\_compression/I | comprmode/I | N.A. | comprmode/I |
| 148 | X\_SPI0\_SCLK | Scan\_ATE\_clk0/I | ATE\_clk/I | MBIST\_ATE\_clk/I | ATE\_clk/I |
| 149 | X\_SPI0\_MOSI | Scan\_enable/I | DFTSE/I | N.A. | DFTSE/I |
| 150 | X\_SPI0\_MISO | OCC\_reset/I | OCC\_reset/I | N.A. | OCC\_reset/I |
| 151 | X\_SPI0\_SS\_N | N.A. | N.A. | N.A. | N.A. |
| 152 | X\_SPI1\_SCLK | Scan\_ATE\_clk1/I | ATCLKEN/I | N.A. | N.A. |
| 153 | X\_SPI1\_MOSI | N.A. | rstdisable/I | N.A. | N.A. |
| 154 | X\_SPI1\_MISO | N.A. | ramhold/I | N.A. | N.A. |
| 155 | X\_SPI1\_SS\_N | N.A. | N.A. | N.A. | N.A. |
| 156 | X\_I2C\_SCL | N.A. | N.A. | N.A. | N.A. |
| 157 | X\_I2C\_SDA | N.A. | N.A. | N.A. | N.A. |
| 158 | X\_UART\_TXD | N.A. | N.A. | N.A. | N.A. |
| 159 | X\_UART\_RXD | N.A. | N.A. | N.A. | N.A. |
| 160 | X\_GPIO\_PA\_0 | scan\_in\_0/I | scan\_in\_0/I | Sram1\_TEST\_H/I | BIST\_EN1/I |
| 161 | X\_GPIO\_PA\_1 | scan\_in\_1/I | scan\_in\_1/I | Sram1\_Fail\_H/O | BIST\_GO1/O |
| 162 | X\_GPIO\_PA\_2 | scan\_in\_2/I | scan\_in\_2/I | Sram1\_Test\_Done/O | BIST\_DONE1/O |
| 163 | X\_GPIO\_PA\_3 | scan\_in\_3/I | scan\_in\_3/I | Sram2\_TEST\_H/I | BIST\_EN2/I |
| 164 | X\_GPIO\_PA\_4 | scan\_in\_4/I | scan\_in\_4/I | Sram2\_Fail\_H/O | BIST\_GO2/O |
| 165 | X\_GPIO\_PA\_5 | scan\_in\_5/I | scan\_in\_5/I | Sram2\_Test\_Done/O | BIST\_DONE2/O |
| 166 | X\_GPIO\_PA\_6 | scan\_in\_6/I | N.A. | Sram3\_TEST\_H/I | BIST\_EN3/I |
| 167 | X\_GPIO\_PA\_7 | scan\_in\_7/I | N.A. | Sram3\_Fail\_H/O | BIST\_GO3/O |
| 168 | X\_GPIO\_PB\_0 | scan\_in\_8/I | BIST\_RESET/I | Sram3\_Test\_Done/O | BIST\_DONE3/O |
| 169 | X\_GPIO\_PB\_1 | scan\_in\_9/I | N.A. | Sram4\_TEST\_H/I | BIST\_EN4/I |
| 170 | X\_GPIO\_PB\_2 | scan\_in\_10/I | N.A. | Sram4\_Fail\_H/O | BIST\_GO4/O |
| 171 | X\_GPIO\_PB\_3 | scan\_in\_11/I | N.A. | Sram4\_Test\_Done/O | BIST\_DONE4/O |
| 172 | X\_GPIO\_PB\_4 | scan\_out\_0/O | scan\_out\_0/O | Sram5\_TEST\_H/I | BIST\_EN5/I |
| 173 | X\_GPIO\_PB\_5 | scan\_out\_1/O | scan\_out\_1/O | Sram5\_Fail\_H/O | BIST\_GO5/O |
| 174 | X\_GPIO\_PB\_6 | scan\_out\_2/O | scan\_out\_2/O | Sram5\_Test\_Done/O | BIST\_DONE5/O |
| 175 | X\_GPIO\_PB\_7 | scan\_out\_3/O | scan\_out\_3/O | Sram6\_TEST\_H/I | BIST\_EN6/I |
| 176 | X\_GPIO\_PC\_0 | scan\_out\_4/O | scan\_out\_4/O | Sram6\_Fail\_H/O | BIST\_GO6/O |
| 177 | X\_GPIO\_PC\_1 | scan\_out\_5/O | scan\_out\_5/O | Sram6\_Test\_Done/O | BIST\_DONE6/O |
| 178 | X\_GPIO\_PC\_2 | scan\_out\_6/O | BIST\_EN1/I | Rom\_TEST\_H/I | N.A. |
| 179 | X\_GPIO\_PC\_3 | scan\_out\_7/O | BIST\_EN2/I | Rom\_MISR\_Clock/I | N.A. |
| 180 | X\_GPIO\_PC\_4 | scan\_out\_8/O | BIST\_EN3/I | Rom\_MISR\_ScanEnable/I | N.A. |
| 181 | X\_GPIO\_PC\_5 | scan\_out\_9/O | BIST\_EN4/I | Rom\_Test\_Done/O | N.A. |
| 182 | X\_GPIO\_PC\_6 | scan\_out\_10/O | BIST\_EN5/I | Rom\_MISR\_ScanOut0/O | N.A. |
| 183 | X\_GPIO\_PC\_7 | scan\_out\_11/O | BIST\_EN6/I | Rom\_MISR\_ScanOut1/O | N.A. |
| 184 | X\_SPI\_NOR\_SCLK | N.A. | N.A. | N.A. | N.A. |
| 185 | X\_SPI\_NOR\_MOSI | N.A. | N.A. | N.A. | N.A. |
| 186 | X\_SPI\_NOR\_MISO | N.A. | N.A. | N.A. | N.A. |
| 187 | X\_SPI\_NOR\_SS\_N | N.A. | N.A. | N.A. | N.A. |
| 188 | X\_SPI\_NOR\_SIO2\_IN | N.A. | N.A. | N.A. | N.A. |
| 189 | X\_SPI\_NOR\_SIO3\_IN | N.A. | N.A. | N.A. | N.A. |

* + 1. Analog Test Mode

Table 52. Analog Test Mode IOMUX Table

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | Common Pin Name | PLL Test MODE  (MODE\_SEL = 4’h6) | RTC Test MODE  (MODE\_SEL = 4’h8) |
| 1 | X\_POR\_N | N.A. |  |
| 2 | X\_SYS\_RST\_N | N.A. |  |
| 3 | X\_SYSPWREN | N.A. |  |
| 4 | X\_WAKE\_IRQ | N.A. |  |
| 5 | X\_MAIN\_CLK\_I | N.A. |  |
| 6 | X\_MAIN\_CLK\_O | N.A. |  |
| 7 | X\_RTC\_SCL | N.A. |  |
| 8 | X\_RTC\_SDA | N.A. |  |
| 9 | X\_RTC\_INT | N.A. |  |
| 10 | X\_RTC\_32K\_CLK | N.A. |  |
| 11 | X\_MODE\_SEL\_0 | N.A. |  |
| 12 | X\_MODE\_SEL\_1 | N.A. |  |
| 13 | X\_MODE\_SEL\_2 | N.A. |  |
| 14 | X\_MODE\_SEL\_3 | N.A. |  |
| 15 | X\_BOOT\_0 | N.A. |  |
| 16 | X\_BOOT\_1 | N.A. |  |
| 17 | X\_DDR\_CLK | N.A. |  |
| 18 | X\_DDR\_CLK\_N | N.A. |  |
| 19 | X\_DDR\_DATA\_0 | N.A. |  |
| 20 | X\_DDR\_DATA\_1 | N.A. |  |
| 21 | X\_DDR\_DATA\_2 | N.A. |  |
| 22 | X\_DDR\_DATA\_3 | N.A. |  |
| 23 | X\_DDR\_DATA\_4 | N.A. |  |
| 24 | X\_DDR\_DATA\_5 | N.A. |  |
| 25 | X\_DDR\_DATA\_6 | N.A. |  |
| 26 | X\_DDR\_DATA\_7 | N.A. |  |
| 27 | X\_DDR\_DATA\_8 | N.A. |  |
| 28 | X\_DDR\_DATA\_9 | N.A. |  |
| 29 | X\_DDR\_DATA\_10 | N.A. |  |
| 30 | X\_DDR\_DATA\_11 | N.A. |  |
| 31 | X\_DDR\_DATA\_12 | N.A. |  |
| 32 | X\_DDR\_DATA\_13 | N.A. |  |
| 33 | X\_DDR\_DATA\_14 | N.A. |  |
| 34 | X\_DDR\_DATA\_15 | N.A. |  |
| 35 | X\_DDR\_DATA\_16 | N.A. |  |
| 36 | X\_DDR\_DATA\_17 | N.A. |  |
| 37 | X\_DDR\_DATA\_18 | N.A. |  |
| 38 | X\_DDR\_DATA\_19 | N.A. |  |
| 39 | X\_DDR\_DATA\_20 | N.A. |  |
| 40 | X\_DDR\_DATA\_21 | N.A. |  |
| 41 | X\_DDR\_DATA\_22 | N.A. |  |
| 42 | X\_DDR\_DATA\_23 | N.A. |  |
| 43 | X\_DDR\_DATA\_24 | N.A. |  |
| 44 | X\_DDR\_DATA\_25 | N.A. |  |
| 45 | X\_DDR\_DATA\_26 | N.A. |  |
| 46 | X\_DDR\_DATA\_27 | N.A. |  |
| 47 | X\_DDR\_DATA\_28 | N.A. |  |
| 48 | X\_DDR\_DATA\_29 | N.A. |  |
| 49 | X\_DDR\_DATA\_30 | N.A. |  |
| 50 | X\_DDR\_DATA\_31 | N.A. |  |
| 51 | X\_DDR\_ADDR\_0 | N.A. |  |
| 52 | X\_DDR\_ADDR\_1 | N.A. |  |
| 53 | X\_DDR\_ADDR\_2 | N.A. |  |
| 54 | X\_DDR\_ADDR\_3 | N.A. |  |
| 55 | X\_DDR\_ADDR\_4 | N.A. |  |
| 56 | X\_DDR\_ADDR\_5 | N.A. |  |
| 57 | X\_DDR\_ADDR\_6 | N.A. |  |
| 58 | X\_DDR\_ADDR\_7 | N.A. |  |
| 59 | X\_DDR\_ADDR\_8 | N.A. |  |
| 60 | X\_DDR\_ADDR\_9 | N.A. |  |
| 61 | X\_DDR \_CKE | N.A. |  |
| 62 | X\_DDR\_CS\_N | N.A. |  |
| 63 | X\_DDR\_DM\_0 | N.A. |  |
| 64 | X\_DDR\_DM\_1 | N.A. |  |
| 65 | X\_DDR\_DM\_2 | N.A. |  |
| 66 | X\_DDR\_DM\_3 | N.A. |  |
| 67 | X\_DDR\_DQS\_M\_0 | N.A. |  |
| 68 | X\_DDR\_DQS\_M\_1 | N.A. |  |
| 69 | X\_DDR\_DQS\_M\_2 | N.A. |  |
| 70 | X\_DDR\_DQS\_M\_3 | N.A. |  |
| 71 | X\_DDR\_DQS\_P\_0 | N.A. |  |
| 72 | X\_DDR\_DQS\_P\_1 | N.A. |  |
| 73 | X\_DDR\_DQS\_P\_2 | N.A. |  |
| 74 | X\_DDR\_DQS\_P\_3 | N.A. |  |
| 75 | X\_DDR\_ODT | N.A. |  |
| 76 | X\_SDIO0\_SDCLK | N.A. |  |
| 77 | X\_SDIO0\_CMD | N.A. |  |
| 78 | X\_SDIO0\_DAT \_0 | N.A. |  |
| 79 | X\_SDIO0\_DAT \_1 | N.A. |  |
| 80 | X\_SDIO0\_DAT\_2 | N.A. |  |
| 81 | X\_SDIO0\_DAT \_3 | N.A. |  |
| 82 | X\_SDIO0\_SDCD\_N | N.A. |  |
| 83 | X\_SDIO0\_SDWP\_N | N.A. |  |
| 84 | X\_SDIO0\_CLE | N.A. |  |
| 85 | X\_SDIO0\_LED | N.A. |  |
| 86 | X\_SDIO0\_BUS\_POW | N.A. |  |
| 87 | X\_SDIO0\_BUS\_VOLT | N.A. |  |
| 88 | X\_SDIO1\_SDCLK | N.A. |  |
| 89 | X\_SDIO1\_CMD | N.A. |  |
| 90 | X\_SDIO1\_DAT\_0 | N.A. |  |
| 91 | X\_SDIO1\_DAT\_1 | N.A. |  |
| 92 | X\_SDIO1\_DAT\_2 | N.A. |  |
| 93 | X\_SDIO1\_DAT\_3 | N.A. |  |
| 94 | X\_SDIO1\_SDCD\_N | N.A. |  |
| 95 | X\_SDIO1\_SDWP\_N | N.A. |  |
| 96 | X\_SDIO1\_CLE | N.A. |  |
| 97 | X\_SDIO1\_LED | N.A. |  |
| 98 | X\_SDIO1\_BUS\_POW | N.A. |  |
| 99 | X\_SDIO1\_BUS\_VOLT | N.A. |  |
| 100 | X\_GMAC\_CLKT | N.A. |  |
| 101 | X\_GMAC\_CLKR | N.A. |  |
| 102 | X\_GMAC\_PHY\_50M | N.A. |  |
| 103 | X\_GMAC\_PHY\_125M | N.A. |  |
| 104 | X\_GMAC\_CLKO | N.A. |  |
| 105 | X\_GMAC\_TXD\_0 | N.A. |  |
| 106 | X\_GMAC\_TXD\_1 | N.A. |  |
| 107 | X\_GMAC\_TXD\_2 | N.A. |  |
| 108 | X\_GMAC\_TXD\_3 | N.A. |  |
| 109 | X\_GMAC\_TXD\_4 | N.A. |  |
| 110 | X\_GMAC\_TXD\_5 | N.A. |  |
| 111 | X\_GMAC\_TXD\_6 | N.A. |  |
| 112 | X\_GMAC\_TXD\_7 | N.A. |  |
| 113 | X\_GMAC\_MDIO | N.A. |  |
| 114 | X\_GMAC\_MDC | N.A. |  |
| 115 | X\_GMAC\_RXER | N.A. |  |
| 116 | X\_GMAC\_RXDV | N.A. |  |
| 117 | X\_GMAC\_COL | N.A. |  |
| 118 | X\_GMAC\_CRS | N.A. |  |
| 119 | X\_GMAC\_RXD\_0 | N.A. |  |
| 120 | X\_GMAC\_RXD\_1 | N.A. |  |
| 121 | X\_GMAC\_RXD\_2 | N.A. |  |
| 122 | X\_GMAC\_RXD\_3 | N.A. |  |
| 123 | X\_GMAC\_RXD\_4 | N.A. |  |
| 124 | X\_GMAC\_RXD\_5 | N.A. |  |
| 125 | X\_GMAC\_RXD\_6 | N.A. |  |
| 126 | X\_GMAC\_RXD\_7 | N.A. |  |
| 127 | X\_GMAC\_TXEN | N.A. |  |
| 128 | X\_GMAC\_TXER | N.A. |  |
| 129 | X\_IVA\_EXTCLK | N.A. |  |
| 130 | X\_IVA\_PIXCLK | N.A. |  |
| 131 | X\_IVA\_FRAME\_VALID | N.A. |  |
| 132 | X\_IVA\_LINE\_VALID | N.A. |  |
| 133 | X\_IVA\_DATA\_0 | N.A. |  |
| 134 | X\_IVA\_DATA\_1 | N.A. |  |
| 135 | X\_IVA\_DATA\_2 | N.A. |  |
| 136 | X\_IVA\_DATA\_3 | N.A. |  |
| 137 | X\_IVA\_DATA\_4 | N.A. |  |
| 138 | X\_IVA\_DATA\_5 | N.A. |  |
| 139 | X\_IVA\_DATA\_6 | N.A. |  |
| 140 | X\_IVA\_DATA\_7 | N.A. |  |
| 141 | X\_IVA\_DATA\_8 | N.A. |  |
| 142 | X\_IVA\_DATA\_9 | N.A. |  |
| 143 | X\_IVA\_DATA\_10 | N.A. |  |
| 144 | X\_IVA\_DATA\_11 | N.A. |  |
| 145 | X\_PWM\_0 | N.A. |  |
| 146 | X\_PWM\_1 | N.A. |  |
| 147 | X\_PWM\_2 | N.A. |  |
| 148 | X\_SPI0\_SCLK | N.A. |  |
| 149 | X\_SPI0\_MOSI | N.A. |  |
| 150 | X\_SPI0\_MISO | N.A. |  |
| 151 | X\_SPI0\_SS\_N | N.A. |  |
| 152 | X\_SPI1\_SCLK | N.A. |  |
| 153 | X\_SPI1\_MOSI | N.A. |  |
| 154 | X\_SPI1\_MISO | N.A. |  |
| 155 | X\_SPI1\_SS\_N | N.A. |  |
| 156 | X\_I2C\_SCL | N.A. |  |
| 157 | X\_I2C\_SDA | N.A. |  |
| 158 | X\_UART\_TXD | N.A. |  |
| 159 | X\_UART\_RXD | N.A. |  |
| 160 | X\_GPIO\_PA\_0 | DIV16\_CPLLOUT\_MAX/O |  |
| 161 | X\_GPIO\_PA\_1 | DIV16\_CPLLOUT\_MIN/O |  |
| 162 | X\_GPIO\_PA\_2 | DIV16\_DPLLOUT\_MAX/O |  |
| 163 | X\_GPIO\_PA\_3 | DIV16\_DPLLOUT\_MIN/O |  |
| 164 | X\_GPIO\_PA\_4 | DIV16\_SPLLOUT\_MAX/O |  |
| 165 | X\_GPIO\_PA\_5 | DIV16\_SPLLOUT\_MIN/O |  |
| 166 | X\_GPIO\_PA\_6 | DIV16\_VPLLOUT\_MAX/O |  |
| 167 | X\_GPIO\_PA\_7 | DIV16\_VPLLOUT\_MIN/O |  |
| 168 | X\_GPIO\_PB\_0 | N.A. |  |
| 169 | X\_GPIO\_PB\_1 | N.A. |  |
| 170 | X\_GPIO\_PB\_2 | N.A. |  |
| 171 | X\_GPIO\_PB\_3 | N.A. |  |
| 172 | X\_GPIO\_PB\_4 | N.A. |  |
| 173 | X\_GPIO\_PB\_5 | N.A. |  |
| 174 | X\_GPIO\_PB\_6 | N.A. |  |
| 175 | X\_GPIO\_PB\_7 | N.A. |  |
| 176 | X\_GPIO\_PC\_0 | N.A. |  |
| 177 | X\_GPIO\_PC\_1 | N.A. |  |
| 178 | X\_GPIO\_PC\_2 | N.A. |  |
| 179 | X\_GPIO\_PC\_3 | N.A. |  |
| 180 | X\_GPIO\_PC\_4 | N.A. |  |
| 181 | X\_GPIO\_PC\_5 | N.A. |  |
| 182 | X\_GPIO\_PC\_6 | N.A. |  |
| 183 | X\_GPIO\_PC\_7 | N.A. |  |
| 184 | X\_SPI\_NOR\_SCLK | N.A. |  |
| 185 | X\_SPI\_NOR\_MOSI | N.A. |  |
| 186 | X\_SPI\_NOR\_MISO | N.A. |  |
| 187 | X\_SPI\_NOR\_SS\_N | N.A. |  |
| 188 | X\_SPI\_NOR\_SIO2\_IN | N.A. |  |
| 189 | X\_SPI\_NOR\_SIO3\_IN | N.A. |  |

* + 1. NAND-Tree Test Mode

Table 53. NAND-Tree Test Mode IOMUX Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Common Pin Name | NAND Tree Test  (MODE\_SEL = 4’h9) | NO. | Common Pin Name | NAND Tree Test  (MODE\_SEL = 4’h9) |
| 1 | X\_POR\_N | nand\_tree\_a0/I | 96 | X\_SDIO1\_CLE | nand\_tree\_a29/I |
| 2 | X\_SYS\_RST\_N | nand\_tree\_out/O | 97 | X\_SDIO1\_LED | nand\_tree\_a30/I |
| 3 | X\_SYSPWREN | nand\_tree\_a1/I | 98 | X\_SDIO1\_BUS\_POW | nand\_tree\_a31/I |
| 4 | X\_WAKE\_IRQ | nand\_tree\_a2/I | 99 | X\_SDIO1\_BUS\_VOLT | nand\_tree\_a32/I |
| 5 | X\_MAIN\_CLK\_I | N.A. | 100 | X\_GMAC\_CLKT | nand\_tree\_a33/I |
| 6 | X\_MAIN\_CLK\_O | N.A. | 101 | X\_GMAC\_CLKR | nand\_tree\_a34/I |
| 7 | X\_RTC\_SCL | nand\_tree\_a3/I | 102 | X\_GMAC\_PHY\_50M | nand\_tree\_a35/I |
| 8 | X\_RTC\_SDA | nand\_tree\_a4/I | 103 | X\_GMAC\_PHY\_125M | nand\_tree\_a36/I |
| 9 | X\_RTC\_INT | nand\_tree\_a5/I | 104 | X\_GMAC\_CLKO | nand\_tree\_a37/I |
| 10 | X\_RTC\_32K\_CLK | nand\_tree\_a6/I | 105 | X\_GMAC\_TXD\_0 | nand\_tree\_a38/I |
| 11 | X\_MODE\_SEL\_0 | N.A. | 106 | X\_GMAC\_TXD\_1 | nand\_tree\_a39/I |
| 12 | X\_MODE\_SEL\_1 | N.A. | 107 | X\_GMAC\_TXD\_2 | nand\_tree\_a40/I |
| 13 | X\_MODE\_SEL\_2 | N.A. | 108 | X\_GMAC\_TXD\_3 | nand\_tree\_a41/I |
| 14 | X\_MODE\_SEL\_3 | N.A. | 109 | X\_GMAC\_TXD\_4 | nand\_tree\_a42/I |
| 15 | X\_BOOT\_0 | nand\_tree\_a7/I | 110 | X\_GMAC\_TXD\_5 | nand\_tree\_a43/I |
| 16 | X\_BOOT\_1 | nand\_tree\_a8/I | 111 | X\_GMAC\_TXD\_6 | nand\_tree\_a44/I |
| 17 | X\_DDR\_CLK | N.A. | 112 | X\_GMAC\_TXD\_7 | nand\_tree\_a45/I |
| 18 | X\_DDR\_CLK\_N | N.A. | 113 | X\_GMAC\_MDIO | nand\_tree\_a46/I |
| 19 | X\_DDR\_DATA\_0 | N.A. | 114 | X\_GMAC\_MDC | nand\_tree\_a47/I |
| 20 | X\_DDR\_DATA\_1 | N.A. | 115 | X\_GMAC\_RXER | nand\_tree\_a48/I |
| 21 | X\_DDR\_DATA\_2 | N.A. | 116 | X\_GMAC\_RXDV | nand\_tree\_a49/I |
| 22 | X\_DDR\_DATA\_3 | N.A. | 117 | X\_GMAC\_COL | nand\_tree\_a50/I |
| 23 | X\_DDR\_DATA\_4 | N.A. | 118 | X\_GMAC\_CRS | nand\_tree\_a51/I |
| 24 | X\_DDR\_DATA\_5 | N.A. | 119 | X\_GMAC\_RXD\_0 | nand\_tree\_a52/I |
| 25 | X\_DDR\_DATA\_6 | N.A. | 120 | X\_GMAC\_RXD\_1 | nand\_tree\_a53/I |
| 26 | X\_DDR\_DATA\_7 | N.A. | 121 | X\_GMAC\_RXD\_2 | nand\_tree\_a54/I |
| 27 | X\_DDR\_DATA\_8 | N.A. | 122 | X\_GMAC\_RXD\_3 | nand\_tree\_a55/I |
| 28 | X\_DDR\_DATA\_9 | N.A. | 123 | X\_GMAC\_RXD\_4 | nand\_tree\_a56/I |
| 29 | X\_DDR\_DATA\_10 | N.A. | 124 | X\_GMAC\_RXD\_5 | nand\_tree\_a57/I |
| 30 | X\_DDR\_DATA\_11 | N.A. | 125 | X\_GMAC\_RXD\_6 | nand\_tree\_a58/I |
| 31 | X\_DDR\_DATA\_12 | N.A. | 126 | X\_GMAC\_RXD\_7 | nand\_tree\_a59/I |
| 32 | X\_DDR\_DATA\_13 | N.A. | 127 | X\_GMAC\_TXEN | nand\_tree\_a60/I |
| 33 | X\_DDR\_DATA\_14 | N.A. | 128 | X\_GMAC\_TXER | nand\_tree\_a61/I |
| 34 | X\_DDR\_DATA\_15 | N.A. | 129 | X\_IVA\_EXTCLK | nand\_tree\_b0/I |
| 35 | X\_DDR\_DATA\_16 | N.A. | 130 | X\_IVA\_PIXCLK | nand\_tree\_b1/I |
| 36 | X\_DDR\_DATA\_17 | N.A. | 131 | X\_IVA\_FRAME\_VALID | nand\_tree\_b2/I |
| 37 | X\_DDR\_DATA\_18 | N.A. | 132 | X\_IVA\_LINE\_VALID | nand\_tree\_b3/I |
| 38 | X\_DDR\_DATA\_19 | N.A. | 133 | X\_IVA\_DATA\_0 | nand\_tree\_b4/I |
| 39 | X\_DDR\_DATA\_20 | N.A. | 134 | X\_IVA\_DATA\_1 | nand\_tree\_b5/I |
| 40 | X\_DDR\_DATA\_21 | N.A. | 135 | X\_IVA\_DATA\_2 | nand\_tree\_b6/I |
| 41 | X\_DDR\_DATA\_22 | N.A. | 136 | X\_IVA\_DATA\_3 | nand\_tree\_b7/I |
| 42 | X\_DDR\_DATA\_23 | N.A. | 137 | X\_IVA\_DATA\_4 | nand\_tree\_b8/I |
| 43 | X\_DDR\_DATA\_24 | N.A. | 138 | X\_IVA\_DATA\_5 | nand\_tree\_b9/I |
| 44 | X\_DDR\_DATA\_25 | N.A. | 139 | X\_IVA\_DATA\_6 | nand\_tree\_b10/I |
| 45 | X\_DDR\_DATA\_26 | N.A. | 140 | X\_IVA\_DATA\_7 | nand\_tree\_b11/I |
| 46 | X\_DDR\_DATA\_27 | N.A. | 141 | X\_IVA\_DATA\_8 | nand\_tree\_b12/I |
| 47 | X\_DDR\_DATA\_28 | N.A. | 142 | X\_IVA\_DATA\_9 | nand\_tree\_b13/I |
| 48 | X\_DDR\_DATA\_29 | N.A. | 143 | X\_IVA\_DATA\_10 | nand\_tree\_b14/I |
| 49 | X\_DDR\_DATA\_30 | N.A. | 144 | X\_IVA\_DATA\_11 | nand\_tree\_b15/I |
| 50 | X\_DDR\_DATA\_31 | N.A. | 145 | X\_PWM\_0 | nand\_tree\_b16/I |
| 51 | X\_DDR\_ADDR\_0 | N.A. | 146 | X\_PWM\_1 | nand\_tree\_b17/I |
| 52 | X\_DDR\_ADDR\_1 | N.A. | 147 | X\_PWM\_2 | nand\_tree\_b18/I |
| 53 | X\_DDR\_ADDR\_2 | N.A. | 148 | X\_SPI0\_SCLK | nand\_tree\_b19/I |
| 54 | X\_DDR\_ADDR\_3 | N.A. | 149 | X\_SPI0\_MOSI | nand\_tree\_b20/I |
| 55 | X\_DDR\_ADDR\_4 | N.A. | 150 | X\_SPI0\_MISO | nand\_tree\_b21/I |
| 56 | X\_DDR\_ADDR\_5 | N.A. | 151 | X\_SPI0\_SS\_N | nand\_tree\_b22/I |
| 57 | X\_DDR\_ADDR\_6 | N.A. | 152 | X\_SPI1\_SCLK | nand\_tree\_b23/I |
| 58 | X\_DDR\_ADDR\_7 | N.A. | 153 | X\_SPI1\_MOSI | nand\_tree\_b24/I |
| 59 | X\_DDR\_ADDR\_8 | N.A. | 154 | X\_SPI1\_MISO | nand\_tree\_b25/I |
| 60 | X\_DDR\_ADDR\_9 | N.A. | 155 | X\_SPI1\_SS\_N | nand\_tree\_b26/I |
| 61 | X\_DDR \_CKE | N.A. | 156 | X\_I2C\_SCL | nand\_tree\_b27/I |
| 62 | X\_DDR\_CS\_N | N.A. | 157 | X\_I2C\_SDA | nand\_tree\_b28/I |
| 63 | X\_DDR\_DM\_0 | N.A. | 158 | X\_UART\_TXD | nand\_tree\_b29/I |
| 64 | X\_DDR\_DM\_1 | N.A. | 159 | X\_UART\_RXD | nand\_tree\_b30/I |
| 65 | X\_DDR\_DM\_2 | N.A. | 160 | X\_GPIO\_PA\_0 | nand\_tree\_b31/I |
| 66 | X\_DDR\_DM\_3 | N.A. | 161 | X\_GPIO\_PA\_1 | nand\_tree\_b32/I |
| 67 | X\_DDR\_DQS\_M\_0 | N.A. | 162 | X\_GPIO\_PA\_2 | nand\_tree\_b33/I |
| 68 | X\_DDR\_DQS\_M\_1 | N.A. | 163 | X\_GPIO\_PA\_3 | nand\_tree\_b34/I |
| 69 | X\_DDR\_DQS\_M\_2 | N.A. | 164 | X\_GPIO\_PA\_4 | nand\_tree\_b35/I |
| 70 | X\_DDR\_DQS\_M\_3 | N.A. | 165 | X\_GPIO\_PA\_5 | nand\_tree\_b36/I |
| 71 | X\_DDR\_DQS\_P\_0 | N.A. | 166 | X\_GPIO\_PA\_6 | nand\_tree\_b37/I |
| 72 | X\_DDR\_DQS\_P\_1 | N.A. | 167 | X\_GPIO\_PA\_7 | nand\_tree\_b38/I |
| 73 | X\_DDR\_DQS\_P\_2 | N.A. | 168 | X\_GPIO\_PB\_0 | nand\_tree\_b39/I |
| 74 | X\_DDR\_DQS\_P\_3 | N.A. | 169 | X\_GPIO\_PB\_1 | nand\_tree\_b40/I |
| 75 | X\_DDR\_ODT | N.A. | 170 | X\_GPIO\_PB\_2 | nand\_tree\_b41/I |
| 76 | X\_SDIO0\_SDCLK | nand\_tree\_a9/I | 171 | X\_GPIO\_PB\_3 | nand\_tree\_b42/I |
| 77 | X\_SDIO0\_CMD | nand\_tree\_a10/I | 172 | X\_GPIO\_PB\_4 | nand\_tree\_b43/I |
| 78 | X\_SDIO0\_DAT \_0 | nand\_tree\_a11/I | 173 | X\_GPIO\_PB\_5 | nand\_tree\_b44/I |
| 79 | X\_SDIO0\_DAT \_1 | nand\_tree\_a12/I | 174 | X\_GPIO\_PB\_6 | nand\_tree\_b45/I |
| 80 | X\_SDIO0\_DAT\_2 | nand\_tree\_a13/I | 175 | X\_GPIO\_PB\_7 | nand\_tree\_b46/I |
| 81 | X\_SDIO0\_DAT \_3 | nand\_tree\_a14/I | 176 | X\_GPIO\_PC\_0 | nand\_tree\_b47/I |
| 82 | X\_SDIO0\_SDCD\_N | nand\_tree\_a15/I | 177 | X\_GPIO\_PC\_1 | nand\_tree\_b48/I |
| 83 | X\_SDIO0\_SDWP\_N | nand\_tree\_a16/I | 178 | X\_GPIO\_PC\_2 | nand\_tree\_b49/I |
| 84 | X\_SDIO0\_CLE | nand\_tree\_a17/I | 179 | X\_GPIO\_PC\_3 | nand\_tree\_b50/I |
| 85 | X\_SDIO0\_LED | nand\_tree\_a18/I | 180 | X\_GPIO\_PC\_4 | nand\_tree\_b51/I |
| 86 | X\_SDIO0\_BUS\_POW | nand\_tree\_a19/I | 181 | X\_GPIO\_PC\_5 | nand\_tree\_b52/I |
| 87 | X\_SDIO0\_BUS\_VOLT | nand\_tree\_a20/I | 182 | X\_GPIO\_PC\_6 | nand\_tree\_b53/I |
| 88 | X\_SDIO1\_SDCLK | nand\_tree\_a21/I | 183 | X\_GPIO\_PC\_7 | nand\_tree\_b54/I |
| 89 | X\_SDIO1\_CMD | nand\_tree\_a22/I | 184 | X\_SPI\_NOR\_SCLK | nand\_tree\_b55/I |
| 90 | X\_SDIO1\_DAT\_0 | nand\_tree\_a23/I | 185 | X\_SPI\_NOR\_MOSI | nand\_tree\_b56/I |
| 91 | X\_SDIO1\_DAT\_1 | nand\_tree\_a24/I | 186 | X\_SPI\_NOR\_MISO | nand\_tree\_b57/I |
| 92 | X\_SDIO1\_DAT\_2 | nand\_tree\_a25/I | 187 | X\_SPI\_NOR\_SS\_N | nand\_tree\_b58/I |
| 93 | X\_SDIO1\_DAT\_3 | nand\_tree\_a26/I | 188 | X\_SPI\_NOR\_SIO2\_IN | nand\_tree\_b59/I |
| 94 | X\_SDIO1\_SDCD\_N | nand\_tree\_a27/I | 189 | X\_SPI\_NOR\_SIO3\_IN | nand\_tree\_b60/I |
| 95 | X\_SDIO1\_SDWP\_N | nand\_tree\_a28/I |  |  |  |

* + 1. Debug Mode

Table 54. Debug Mode MUX Table (1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO. | Common Pin Name | Engineering Mode | | | | | |
| MODE\_SEL = 4’h1 | | | | | |
| Debug Mode | | | | | |
| reg\_mux\_sel = 4’h1 | reg\_mux\_sel = 4’h2 | reg\_mux\_sel = 4’h3 | reg\_mux\_sel = 4’h4 | reg\_mux\_sel = 4’h5 | reg\_mux\_sel = 4’h6 |
| DDR\_DFI | DDR\_PAD | SDIO | GMAC | ISRAM | DMA |
| 1 | X\_POR\_N | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I |
| 2 | X\_SYS\_RST\_N | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O |
| 3 | X\_SYSPWREN | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O |
| 4 | X\_WAKE\_IRQ | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I |
| 5 | X\_MAIN\_CLK\_I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I |
| 6 | X\_MAIN\_CLK\_O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O |
| 7 | X\_RTC\_SCL | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B |
| 8 | X\_RTC\_SDA | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B |
| 9 | X\_RTC\_INT | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I |
| 10 | X\_RTC\_32K\_CLK | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I |
| 11 | X\_MODE\_SEL\_0 | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I |
| 12 | X\_MODE\_SEL\_1 | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I |
| 13 | X\_MODE\_SEL\_2 | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I |
| 14 | X\_MODE\_SEL\_3 | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I |
| 15 | X\_BOOT\_0 | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I |
| 16 | X\_BOOT\_1 | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I |
| 17 | X\_DDR\_CLK | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O |
| 18 | X\_DDR\_CLK\_N | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O |
| 19 | X\_DDR\_DATA\_0 | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B |
| 20 | X\_DDR\_DATA\_1 | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B |
| 21 | X\_DDR\_DATA\_2 | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B |
| 22 | X\_DDR\_DATA\_3 | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B |
| 23 | X\_DDR\_DATA\_4 | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B |
| 24 | X\_DDR\_DATA\_5 | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B |
| 25 | X\_DDR\_DATA\_6 | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B |
| 26 | X\_DDR\_DATA\_7 | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B |
| 27 | X\_DDR\_DATA\_8 | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B |
| 28 | X\_DDR\_DATA\_9 | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B |
| 29 | X\_DDR\_DATA\_10 | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B |
| 30 | X\_DDR\_DATA\_11 | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B |
| 31 | X\_DDR\_DATA\_12 | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B |
| 32 | X\_DDR\_DATA\_13 | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B |
| 33 | X\_DDR\_DATA\_14 | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B |
| 34 | X\_DDR\_DATA\_15 | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B |
| 35 | X\_DDR\_DATA\_16 | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B |
| 36 | X\_DDR\_DATA\_17 | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B |
| 37 | X\_DDR\_DATA\_18 | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B |
| 38 | X\_DDR\_DATA\_19 | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B |
| 39 | X\_DDR\_DATA\_20 | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B |
| 40 | X\_DDR\_DATA\_21 | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B |
| 41 | X\_DDR\_DATA\_22 | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B |
| 42 | X\_DDR\_DATA\_23 | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B |
| 43 | X\_DDR\_DATA\_24 | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B |
| 44 | X\_DDR\_DATA\_25 | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B |
| 45 | X\_DDR\_DATA\_26 | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B |
| 46 | X\_DDR\_DATA\_27 | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B |
| 47 | X\_DDR\_DATA\_28 | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B |
| 48 | X\_DDR\_DATA\_29 | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B |
| 49 | X\_DDR\_DATA\_30 | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B |
| 50 | X\_DDR\_DATA\_31 | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B |
| 51 | X\_DDR\_ADDR\_0 | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O |
| 52 | X\_DDR\_ADDR\_1 | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O |
| 53 | X\_DDR\_ADDR\_2 | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O |
| 54 | X\_DDR\_ADDR\_3 | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O |
| 55 | X\_DDR\_ADDR\_4 | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O |
| 56 | X\_DDR\_ADDR\_5 | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O |
| 57 | X\_DDR\_ADDR\_6 | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O |
| 58 | X\_DDR\_ADDR\_7 | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O |
| 59 | X\_DDR\_ADDR\_8 | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O |
| 60 | X\_DDR\_ADDR\_9 | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O |
| 61 | X\_DDR \_CKE | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O |
| 62 | X\_DDR\_CS\_N | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O |
| 63 | X\_DDR\_DM\_0 | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B |
| 64 | X\_DDR\_DM\_1 | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B |
| 65 | X\_DDR\_DM\_2 | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B |
| 66 | X\_DDR\_DM\_3 | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B |
| 67 | X\_DDR\_DQS\_M\_0 | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B |
| 68 | X\_DDR\_DQS\_M\_1 | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B |
| 69 | X\_DDR\_DQS\_M\_2 | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B |
| 70 | X\_DDR\_DQS\_M\_3 | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B |
| 71 | X\_DDR\_DQS\_P\_0 | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B |
| 72 | X\_DDR\_DQS\_P\_1 | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B |
| 73 | X\_DDR\_DQS\_P\_2 | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B |
| 74 | X\_DDR\_DQS\_P\_3 | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B |
| 75 | X\_DDR\_ODT | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O |
| 76 | X\_SDIO0\_SDCLK | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O |
| 77 | X\_SDIO0\_CMD | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B |
| 78 | X\_SDIO0\_DAT \_0 | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B |
| 79 | X\_SDIO0\_DAT \_1 | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B |
| 80 | X\_SDIO0\_DAT\_2 | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B |
| 81 | X\_SDIO0\_DAT \_3 | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B |
| 82 | X\_SDIO0\_SDCD\_N | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I |
| 83 | X\_SDIO0\_SDWP\_N | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I |
| 84 | X\_SDIO0\_CLE | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I |
| 85 | X\_SDIO0\_LED | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O |
| 86 | X\_SDIO0\_BUS\_POW | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O |
| 87 | X\_SDIO0\_BUS\_VOLT | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O |
| 88 | X\_SDIO1\_SDCLK | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O |
| 89 | X\_SDIO1\_CMD | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B |
| 90 | X\_SDIO1\_DAT\_0 | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B |
| 91 | X\_SDIO1\_DAT\_1 | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B |
| 92 | X\_SDIO1\_DAT\_2 | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B |
| 93 | X\_SDIO1\_DAT\_3 | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B |
| 94 | X\_SDIO1\_SDCD\_N | DDR\_DBGSIG\_A\_33/O | DDR\_DBGSIG\_B\_33/O | SDIO1\_SDCD\_N /I | sram\_state\_4/O | N.A. | DMA\_DBGSIG\_33/O |
| 95 | X\_SDIO1\_SDWP\_N | DDR\_DBGSIG\_A\_34/O | DDR\_DBGSIG\_B\_34/O | SDIO1\_SDWP\_N /I | sram\_state\_5/O | N.A. | DMA\_DBGSIG\_34/O |
| 96 | X\_SDIO1\_CLE | DDR\_DBGSIG\_A\_35/O | DDR\_DBGSIG\_B\_35/O | SDIO1\_CLE/I | sram\_state\_6/O | N.A. | DMA\_DBGSIG\_35/O |
| 97 | X\_SDIO1\_LED | DDR\_DBGSIG\_A\_36/O | DDR\_DBGSIG\_B\_36/O | SDIO1\_LED/O | sram\_state\_7/O | N.A. | N.A. |
| 98 | X\_SDIO1\_BUS\_POW | DDR\_DBGSIG\_A\_37/O | DDR\_DBGSIG\_B\_37/O | SDIO1\_BUS\_POW/O | N.A. | N.A. | N.A. |
| 99 | X\_SDIO1\_BUS\_VOLT | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O |
| 100 | X\_GMAC\_CLKT | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I |
| 101 | X\_GMAC\_CLKR | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I |
| 102 | X\_GMAC\_PHY\_50M | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I |
| 103 | X\_GMAC\_PHY\_125M | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I |
| 104 | X\_GMAC\_CLKO | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O |
| 105 | X\_GMAC\_TXD\_0 | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O |
| 106 | X\_GMAC\_TXD\_1 | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O |
| 107 | X\_GMAC\_TXD\_2 | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O |
| 108 | X\_GMAC\_TXD\_3 | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O |
| 109 | X\_GMAC\_TXD\_4 | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O |
| 110 | X\_GMAC\_TXD\_5 | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O |
| 111 | X\_GMAC\_TXD\_6 | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O |
| 112 | X\_GMAC\_TXD\_7 | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O |
| 113 | X\_GMAC\_MDIO | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B |
| 114 | X\_GMAC\_MDC | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O |
| 115 | X\_GMAC\_RXER | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I |
| 116 | X\_GMAC\_RXDV | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I |
| 117 | X\_GMAC\_COL | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I |
| 118 | X\_GMAC\_CRS | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I |
| 119 | X\_GMAC\_RXD\_0 | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I |
| 120 | X\_GMAC\_RXD\_1 | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I |
| 121 | X\_GMAC\_RXD\_2 | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I |
| 122 | X\_GMAC\_RXD\_3 | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I |
| 123 | X\_GMAC\_RXD\_4 | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I |
| 124 | X\_GMAC\_RXD\_5 | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I |
| 125 | X\_GMAC\_RXD\_6 | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I |
| 126 | X\_GMAC\_RXD\_7 | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I |
| 127 | X\_GMAC\_TXEN | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O |
| 128 | X\_GMAC\_TXER | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O |
| 129 | X\_IVA\_EXTCLK | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O |
| 130 | X\_IVA\_PIXCLK | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I |
| 131 | X\_IVA\_FRAME\_VALID | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I |
| 132 | X\_IVA\_LINE\_VALID | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I |
| 133 | X\_IVA\_DATA\_0 | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I |
| 134 | X\_IVA\_DATA\_1 | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I |
| 135 | X\_IVA\_DATA\_2 | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I |
| 136 | X\_IVA\_DATA\_3 | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I |
| 137 | X\_IVA\_DATA\_4 | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I |
| 138 | X\_IVA\_DATA\_5 | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I |
| 139 | X\_IVA\_DATA\_6 | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I |
| 140 | X\_IVA\_DATA\_7 | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I |
| 141 | X\_IVA\_DATA\_8 | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I |
| 142 | X\_IVA\_DATA\_9 | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I |
| 143 | X\_IVA\_DATA\_10 | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I |
| 144 | X\_IVA\_DATA\_11 | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I |
| 145 | X\_PWM\_0 | DDR\_DBGSIG\_A\_0/O | DDR\_DBGSIG\_B\_0/O | SDIO\_DBGSIG\_0/O | test\_o\_0/O | ISRAM\_DBGSIG\_0/O | DMA\_DBGSIG\_0/O |
| 146 | X\_PWM\_1 | DDR\_DBGSIG\_A\_1/O | DDR\_DBGSIG\_B\_1/O | SDIO\_DBGSIG\_1/O | test\_o\_1/O | ISRAM\_DBGSIG\_1/O | DMA\_DBGSIG\_1/O |
| 147 | X\_PWM\_2 | DDR\_DBGSIG\_A\_2/O | DDR\_DBGSIG\_B\_2/O | SDIO\_DBGSIG\_2/O | test\_o\_2/O | ISRAM\_DBGSIG\_2/O | DMA\_DBGSIG\_2/O |
| 148 | X\_SPI0\_SCLK | DDR\_DBGSIG\_A\_3/O | DDR\_DBGSIG\_B\_3/O | SDIO\_DBGSIG\_3/O | test\_o\_3/O | ISRAM\_DBGSIG\_3/O | DMA\_DBGSIG\_3/O |
| 149 | X\_SPI0\_MOSI | DDR\_DBGSIG\_A\_4/O | DDR\_DBGSIG\_B\_4/O | SDIO\_DBGSIG\_4/O | test\_o\_4/O | ISRAM\_DBGSIG\_4/O | DMA\_DBGSIG\_4/O |
| 150 | X\_SPI0\_MISO | DDR\_DBGSIG\_A\_5/O | DDR\_DBGSIG\_B\_5/O | SDIO\_DBGSIG\_5/O | test\_o\_5/O | ISRAM\_DBGSIG\_5/O | DMA\_DBGSIG\_5/O |
| 151 | X\_SPI0\_SS\_N | DDR\_DBGSIG\_A\_6/O | DDR\_DBGSIG\_B\_6/O | SDIO\_DBGSIG\_6/O | test\_o\_6/O | ISRAM\_DBGSIG\_6/O | DMA\_DBGSIG\_6/O |
| 152 | X\_SPI1\_SCLK | DDR\_DBGSIG\_A\_7/O | DDR\_DBGSIG\_B\_7/O | SDIO\_DBGSIG\_7/O | test\_o\_7/O | ISRAM\_DBGSIG\_7/O | DMA\_DBGSIG\_7/O |
| 153 | X\_SPI1\_MOSI | DDR\_DBGSIG\_A\_8/O | DDR\_DBGSIG\_B\_8/O | SDIO\_DBGSIG\_8/O | speed\_0/O | ISRAM\_DBGSIG\_8/O | DMA\_DBGSIG\_8/O |
| 154 | X\_SPI1\_MISO | DDR\_DBGSIG\_A\_9/O | DDR\_DBGSIG\_B\_9/O | SDIO\_DBGSIG\_9/O | speed\_1/O | ISRAM\_DBGSIG\_9/O | DMA\_DBGSIG\_9/O |
| 155 | X\_SPI1\_SS\_N | DDR\_DBGSIG\_A\_10/O | DDR\_DBGSIG\_B\_10/O | SDIO\_DBGSIG\_10/O | rsmd\_0/O | ISRAM\_DBGSIG\_10/O | DMA\_DBGSIG\_10/O |
| 156 | X\_I2C\_SCL | DDR\_DBGSIG\_A\_11/O | DDR\_DBGSIG\_B\_11/O | SDIO\_DBGSIG\_11/O | rsmd\_1/O | ISRAM\_DBGSIG\_11/O | DMA\_DBGSIG\_11/O |
| 157 | X\_I2C\_SDA | DDR\_DBGSIG\_A\_12/O | DDR\_DBGSIG\_B\_12/O | SDIO\_DBGSIG\_12/O | rsmd\_2/O | ISRAM\_DBGSIG\_12/O | DMA\_DBGSIG\_12/O |
| 158 | X\_UART\_TXD | DDR\_DBGSIG\_A\_13/O | DDR\_DBGSIG\_B\_13/O | SDIO\_DBGSIG\_13/O | tsmd\_0/O | ISRAM\_DBGSIG\_13/O | DMA\_DBGSIG\_13/O |
| 159 | X\_UART\_RXD | DDR\_DBGSIG\_A\_14/O | DDR\_DBGSIG\_B\_14/O | SDIO\_DBGSIG\_14/O | tsmd\_1/O | ISRAM\_DBGSIG\_14/O | DMA\_DBGSIG\_14/O |
| 160 | X\_GPIO\_PA\_0 | DDR\_DBGSIG\_A\_15/O | DDR\_DBGSIG\_B\_15/O | SDIO\_DBGSIG\_15/O | tsmd\_2/O | ISRAM\_DBGSIG\_15/O | DMA\_DBGSIG\_15/O |
| 161 | X\_GPIO\_PA\_1 | DDR\_DBGSIG\_A\_16/O | DDR\_DBGSIG\_B\_16/O | SDIO\_DBGSIG\_16/O | tsmd\_3/O | ISRAM\_DBGSIG\_16/O | DMA\_DBGSIG\_16/O |
| 162 | X\_GPIO\_PA\_2 | DDR\_DBGSIG\_A\_17/O | DDR\_DBGSIG\_B\_17/O | SDIO\_DBGSIG\_17/O | rlsmd\_0/O | ISRAM\_DBGSIG\_17/O | DMA\_DBGSIG\_17/O |
| 163 | X\_GPIO\_PA\_3 | DDR\_DBGSIG\_A\_18/O | DDR\_DBGSIG\_B\_18/O | SDIO\_DBGSIG\_18/O | rlsmd\_1/O | ISRAM\_DBGSIG\_18/O | DMA\_DBGSIG\_18/O |
| 164 | X\_GPIO\_PA\_4 | DDR\_DBGSIG\_A\_19/O | DDR\_DBGSIG\_B\_19/O | SDIO\_DBGSIG\_19/O | rlsmd\_2/O | ISRAM\_DBGSIG\_19/O | DMA\_DBGSIG\_19/O |
| 165 | X\_GPIO\_PA\_5 | DDR\_DBGSIG\_A\_20/O | DDR\_DBGSIG\_B\_20/O | SDIO\_DBGSIG\_20/O | rlsmd\_3/O | ISRAM\_DBGSIG\_20/O | DMA\_DBGSIG\_20/O |
| 166 | X\_GPIO\_PA\_6 | DDR\_DBGSIG\_A\_21/O | DDR\_DBGSIG\_B\_21/O | SDIO\_DBGSIG\_21/O | tlsmd\_0/O | ISRAM\_DBGSIG\_21/O | DMA\_DBGSIG\_21/O |
| 167 | X\_GPIO\_PA\_7 | DDR\_DBGSIG\_A\_22/O | DDR\_DBGSIG\_B\_22/O | SDIO\_DBGSIG\_22/O | tlsmd\_1/O | ISRAM\_DBGSIG\_22/O | DMA\_DBGSIG\_22/O |
| 168 | X\_GPIO\_PB\_0 | DDR\_DBGSIG\_A\_23/O | DDR\_DBGSIG\_B\_23/O | SDIO\_DBGSIG\_23/O | tlsmd\_2/O | ISRAM\_DBGSIG\_23/O | DMA\_DBGSIG\_23/O |
| 169 | X\_GPIO\_PB\_1 | DDR\_DBGSIG\_A\_24/O | DDR\_DBGSIG\_B\_24/O | SDIO\_DBGSIG\_24/O | tlsmd\_3/O | ISRAM\_DBGSIG\_24/O | DMA\_DBGSIG\_24/O |
| 170 | X\_GPIO\_PB\_2 | DDR\_DBGSIG\_A\_25/O | DDR\_DBGSIG\_B\_25/O | SDIO\_DBGSIG\_25/O | discard/O | ISRAM\_DBGSIG\_25/O | DMA\_DBGSIG\_25/O |
| 171 | X\_GPIO\_PB\_3 | DDR\_DBGSIG\_A\_26/O | DDR\_DBGSIG\_B\_26/O | SDIO\_DBGSIG\_26/O | dma\_mflag\_0/O | ISRAM\_DBGSIG\_26/O | DMA\_DBGSIG\_26/O |
| 172 | X\_GPIO\_PB\_4 | DDR\_DBGSIG\_A\_27/O | DDR\_DBGSIG\_B\_27/O | SDIO\_DBGSIG\_27/O | dma\_mflag\_1/O | N.A. | DMA\_DBGSIG\_27/O |
| 173 | X\_GPIO\_PB\_5 | DDR\_DBGSIG\_A\_28/O | DDR\_DBGSIG\_B\_28/O | SDIO\_DBGSIG\_28/O | dma\_mflag\_2/O | N.A. | DMA\_DBGSIG\_28/O |
| 174 | X\_GPIO\_PB\_6 | DDR\_DBGSIG\_A\_29/O | DDR\_DBGSIG\_B\_29/O | SDIO\_DBGSIG\_29/O | sram\_state\_0/O | N.A. | DMA\_DBGSIG\_29/O |
| 175 | X\_GPIO\_PB\_7 | DDR\_DBGSIG\_A\_30/O | DDR\_DBGSIG\_B\_30/O | SDIO\_DBGSIG\_30/O | sram\_state\_1/O | N.A. | DMA\_DBGSIG\_30/O |
| 176 | X\_GPIO\_PC\_0 | DDR\_DBGSIG\_A\_31/O | DDR\_DBGSIG\_B\_31/O | SDIO\_DBGSIG\_31/O | sram\_state\_2/O | N.A. | DMA\_DBGSIG\_31/O |
| 177 | X\_GPIO\_PC\_1 | DDR\_DBGSIG\_A\_32/O | DDR\_DBGSIG\_B\_32/O | SDIO\_DBGSIG\_32/O | sram\_state\_3/O | N.A. | DMA\_DBGSIG\_32/O |
| 178 | X\_GPIO\_PC\_2 | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I |
| 179 | X\_GPIO\_PC\_3 | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I |
| 180 | X\_GPIO\_PC\_4 | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B |
| 181 | X\_GPIO\_PC\_5 | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I |
| 182 | X\_GPIO\_PC\_6 | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O |
| 183 | X\_GPIO\_PC\_7 | DDR\_DBGSIG\_A\_38/O | DDR\_DBGSIG\_B\_38/O | SDIO\_DBGSIG\_33/O | N.A. | N.A. | N.A. |
| 184 | X\_SPI\_NOR\_SCLK | DDR\_DBGSIG\_A\_39/O | DDR\_DBGSIG\_B\_39/O | SDIO\_DBGSIG\_34/O | N.A. | N.A. | N.A. |
| 185 | X\_SPI\_NOR\_MOSI | DDR\_DBGSIG\_A\_40/O | DDR\_DBGSIG\_B\_40/O | SDIO\_DBGSIG\_35/O | N.A. | N.A. | N.A. |
| 186 | X\_SPI\_NOR\_MISO | N.A. | N.A. | SDIO\_DBGSIG\_36/O | N.A. | N.A. | N.A. |
| 187 | X\_SPI\_NOR\_SS\_N | N.A. | N.A. | SDIO\_DBGSIG\_37/O | N.A. | N.A. | N.A. |
| 188 | X\_SPI\_NOR\_SIO2\_IN | N.A. | N.A. | SDIO\_DBGSIG\_38/O | N.A. | N.A. | N.A. |
| 189 | X\_SPI\_NOR\_SIO3\_IN | N.A. | N.A. | SDIO\_DBGSIG\_39/O | N.A. | N.A. | N.A. |

Table 55. Debug Mode MUX Table (2)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO. | Common Pin Name | Engineering Mode | | | | | |
| MODE\_SEL = 4’h1 | | | | | |
| Debug Mode | | | | | |
| reg\_mux\_sel = 4’h7 | reg\_mux\_sel = 4’h8 | reg\_mux\_sel = 4’h9 | reg\_mux\_sel = 4’ha | reg\_mux\_sel = 4’hb | reg\_mux\_sel = 4’hc |
| CoreSight | SPI | UART | I2C | SPI-NOR | IVA |
| 1 | X\_POR\_N | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I | POR\_N/I |
| 2 | X\_SYS\_RST\_N | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O | SYS\_RST\_N/O |
| 3 | X\_SYSPWREN | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O | SYSPWREN/O |
| 4 | X\_WAKE\_IRQ | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I | WAKE\_IRQ/I |
| 5 | X\_MAIN\_CLK\_I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I | MAIN\_CLK\_I/I |
| 6 | X\_MAIN\_CLK\_O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O | MAIN\_CLK\_O/O |
| 7 | X\_RTC\_SCL | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B | RTC\_SCL/B |
| 8 | X\_RTC\_SDA | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B | RTC\_SDA/B |
| 9 | X\_RTC\_INT | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I | RTC\_INT/I |
| 10 | X\_RTC\_32K\_CLK | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I | RTC\_32K\_CLK/I |
| 11 | X\_MODE\_SEL\_0 | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I | MODE\_SEL\_0/I |
| 12 | X\_MODE\_SEL\_1 | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I | MODE\_SEL\_1/I |
| 13 | X\_MODE\_SEL\_2 | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I | MODE\_SEL\_2/I |
| 14 | X\_MODE\_SEL\_3 | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I | MODE\_SEL\_3/I |
| 15 | X\_BOOT\_0 | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I | BOOT\_0/I |
| 16 | X\_BOOT\_1 | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I | BOOT\_1/I |
| 17 | X\_DDR\_CLK | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O | DDR\_CLK/O |
| 18 | X\_DDR\_CLK\_N | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O | DDR\_CLK\_N/O |
| 19 | X\_DDR\_DATA\_0 | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B | DDR\_DATA\_0/B |
| 20 | X\_DDR\_DATA\_1 | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B | DDR\_DATA\_1/B |
| 21 | X\_DDR\_DATA\_2 | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B | DDR\_DATA\_2/B |
| 22 | X\_DDR\_DATA\_3 | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B | DDR\_DATA\_3/B |
| 23 | X\_DDR\_DATA\_4 | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B | DDR\_DATA\_4/B |
| 24 | X\_DDR\_DATA\_5 | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B | DDR\_DATA\_5/B |
| 25 | X\_DDR\_DATA\_6 | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B | DDR\_DATA\_6/B |
| 26 | X\_DDR\_DATA\_7 | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B | DDR\_DATA\_7/B |
| 27 | X\_DDR\_DATA\_8 | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B | DDR\_DATA\_8/B |
| 28 | X\_DDR\_DATA\_9 | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B | DDR\_DATA\_9/B |
| 29 | X\_DDR\_DATA\_10 | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B | DDR\_DATA\_10/B |
| 30 | X\_DDR\_DATA\_11 | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B | DDR\_DATA\_11/B |
| 31 | X\_DDR\_DATA\_12 | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B | DDR\_DATA\_12/B |
| 32 | X\_DDR\_DATA\_13 | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B | DDR\_DATA\_13/B |
| 33 | X\_DDR\_DATA\_14 | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B | DDR\_DATA\_14/B |
| 34 | X\_DDR\_DATA\_15 | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B | DDR\_DATA\_15/B |
| 35 | X\_DDR\_DATA\_16 | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B | DDR\_DATA\_16/B |
| 36 | X\_DDR\_DATA\_17 | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B | DDR\_DATA\_17/B |
| 37 | X\_DDR\_DATA\_18 | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B | DDR\_DATA\_18/B |
| 38 | X\_DDR\_DATA\_19 | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B | DDR\_DATA\_19/B |
| 39 | X\_DDR\_DATA\_20 | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B | DDR\_DATA\_20/B |
| 40 | X\_DDR\_DATA\_21 | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B | DDR\_DATA\_21/B |
| 41 | X\_DDR\_DATA\_22 | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B | DDR\_DATA\_22/B |
| 42 | X\_DDR\_DATA\_23 | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B | DDR\_DATA\_23/B |
| 43 | X\_DDR\_DATA\_24 | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B | DDR\_DATA\_24/B |
| 44 | X\_DDR\_DATA\_25 | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B | DDR\_DATA\_25/B |
| 45 | X\_DDR\_DATA\_26 | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B | DDR\_DATA\_26/B |
| 46 | X\_DDR\_DATA\_27 | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B | DDR\_DATA\_27/B |
| 47 | X\_DDR\_DATA\_28 | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B | DDR\_DATA\_28/B |
| 48 | X\_DDR\_DATA\_29 | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B | DDR\_DATA\_29/B |
| 49 | X\_DDR\_DATA\_30 | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B | DDR\_DATA\_30/B |
| 50 | X\_DDR\_DATA\_31 | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B | DDR\_DATA\_31/B |
| 51 | X\_DDR\_ADDR\_0 | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O | DDR\_ADDR\_0/O |
| 52 | X\_DDR\_ADDR\_1 | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O | DDR\_ADDR\_1/O |
| 53 | X\_DDR\_ADDR\_2 | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O | DDR\_ADDR\_2/O |
| 54 | X\_DDR\_ADDR\_3 | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O | DDR\_ADDR\_3/O |
| 55 | X\_DDR\_ADDR\_4 | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O | DDR\_ADDR\_4/O |
| 56 | X\_DDR\_ADDR\_5 | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O | DDR\_ADDR\_5/O |
| 57 | X\_DDR\_ADDR\_6 | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O | DDR\_ADDR\_6/O |
| 58 | X\_DDR\_ADDR\_7 | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O | DDR\_ADDR\_7/O |
| 59 | X\_DDR\_ADDR\_8 | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O | DDR\_ADDR\_8/O |
| 60 | X\_DDR\_ADDR\_9 | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O | DDR\_ADDR\_9/O |
| 61 | X\_DDR \_CKE | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O | DDR \_CKE/O |
| 62 | X\_DDR\_CS\_N | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O | DDR\_CS\_N/O |
| 63 | X\_DDR\_DM\_0 | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B | DDR\_DM\_0/B |
| 64 | X\_DDR\_DM\_1 | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B | DDR\_DM\_1/B |
| 65 | X\_DDR\_DM\_2 | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B | DDR\_DM\_2/B |
| 66 | X\_DDR\_DM\_3 | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B | DDR\_DM\_3/B |
| 67 | X\_DDR\_DQS\_M\_0 | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B | DDR\_DQS\_M\_0/B |
| 68 | X\_DDR\_DQS\_M\_1 | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B | DDR\_DQS\_M\_1/B |
| 69 | X\_DDR\_DQS\_M\_2 | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B | DDR\_DQS\_M\_2/B |
| 70 | X\_DDR\_DQS\_M\_3 | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B | DDR\_DQS\_M\_3/B |
| 71 | X\_DDR\_DQS\_P\_0 | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B | DDR\_DQS\_P\_0/B |
| 72 | X\_DDR\_DQS\_P\_1 | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B | DDR\_DQS\_P\_1/B |
| 73 | X\_DDR\_DQS\_P\_2 | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B | DDR\_DQS\_P\_2/B |
| 74 | X\_DDR\_DQS\_P\_3 | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B | DDR\_DQS\_P\_3/B |
| 75 | X\_DDR\_ODT | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O | DDR\_ODT/O |
| 76 | X\_SDIO0\_SDCLK | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O | SDIO0\_SDCLK/O |
| 77 | X\_SDIO0\_CMD | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B | SDIO0\_CMD/B |
| 78 | X\_SDIO0\_DAT \_0 | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B | SDIO0\_DAT\_0/B |
| 79 | X\_SDIO0\_DAT \_1 | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B | SDIO0\_DAT\_1/B |
| 80 | X\_SDIO0\_DAT\_2 | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B | SDIO0\_DAT\_2/B |
| 81 | X\_SDIO0\_DAT \_3 | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B | SDIO0\_DAT\_3/B |
| 82 | X\_SDIO0\_SDCD\_N | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I | SDIO0\_SDCD\_N /I |
| 83 | X\_SDIO0\_SDWP\_N | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I | SDIO0\_SDWP\_N /I |
| 84 | X\_SDIO0\_CLE | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I | SDIO0\_CLE/I |
| 85 | X\_SDIO0\_LED | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O | SDIO0\_LED/O |
| 86 | X\_SDIO0\_BUS\_POW | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O | SDIO0\_BUS\_POW/O |
| 87 | X\_SDIO0\_BUS\_VOLT | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O | SDIO0\_BUS\_VOLT/O |
| 88 | X\_SDIO1\_SDCLK | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O | SDIO1\_SDCLK/O |
| 89 | X\_SDIO1\_CMD | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B | SDIO1\_CMD/B |
| 90 | X\_SDIO1\_DAT\_0 | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B | SDIO1\_DAT\_0/B |
| 91 | X\_SDIO1\_DAT\_1 | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B | SDIO1\_DAT\_1/B |
| 92 | X\_SDIO1\_DAT\_2 | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B | SDIO1\_DAT\_2/B |
| 93 | X\_SDIO1\_DAT\_3 | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B | SDIO1\_DAT\_3/B |
| 94 | X\_SDIO1\_SDCD\_N | N.A. | DBG\_SPI\_OUT\_25/O | DBG\_UART\_OUT\_31/O | DBG\_I2C\_OUT\_31/O | DBG\_SPINOR\_DBGSIG\_33/O | N.A. |
| 95 | X\_SDIO1\_SDWP\_N | N.A. | DBG\_SPI\_OUT\_26/O | N.A. | N.A. | DBG\_SPINOR\_DBGSIG\_34/O | N.A. |
| 96 | X\_SDIO1\_CLE | N.A. | DBG\_SPI\_OUT\_27/O | N.A. | N.A. | DBG\_SPINOR\_DBGSIG\_35/O | N.A. |
| 97 | X\_SDIO1\_LED | N.A. | DBG\_SPI\_OUT\_28/O | N.A. | N.A. | DBG\_SPINOR\_DBGSIG\_36/O | N.A. |
| 98 | X\_SDIO1\_BUS\_POW | N.A. | DBG\_SPI\_OUT\_29/O | N.A. | N.A. | DBG\_SPINOR\_DBGSIG\_37/O | N.A. |
| 99 | X\_SDIO1\_BUS\_VOLT | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O | SDIO1\_BUS\_VOLT/O |
| 100 | X\_GMAC\_CLKT | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I | GMAC\_CLKT/I |
| 101 | X\_GMAC\_CLKR | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I | GMAC\_CLKR/I |
| 102 | X\_GMAC\_PHY\_50M | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I | GMAC\_PHY\_50M/I |
| 103 | X\_GMAC\_PHY\_125M | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I | GMAC\_PHY\_125M/I |
| 104 | X\_GMAC\_CLKO | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O | GMAC\_CLKO/O |
| 105 | X\_GMAC\_TXD\_0 | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O | GMAC\_TXD\_0/O |
| 106 | X\_GMAC\_TXD\_1 | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O | GMAC\_TXD\_1/O |
| 107 | X\_GMAC\_TXD\_2 | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O | GMAC\_TXD\_2/O |
| 108 | X\_GMAC\_TXD\_3 | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O | GMAC\_TXD\_3/O |
| 109 | X\_GMAC\_TXD\_4 | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O | GMAC\_TXD\_4/O |
| 110 | X\_GMAC\_TXD\_5 | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O | GMAC\_TXD\_5/O |
| 111 | X\_GMAC\_TXD\_6 | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O | GMAC\_TXD\_6/O |
| 112 | X\_GMAC\_TXD\_7 | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O | GMAC\_TXD\_7/O |
| 113 | X\_GMAC\_MDIO | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B | GMAC\_MDIO/B |
| 114 | X\_GMAC\_MDC | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O | GMAC\_MDC/O |
| 115 | X\_GMAC\_RXER | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I | GMAC\_RXER/I |
| 116 | X\_GMAC\_RXDV | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I | GMAC\_RXDV/I |
| 117 | X\_GMAC\_COL | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I | GMAC\_COL/I |
| 118 | X\_GMAC\_CRS | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I | GMAC\_CRS/I |
| 119 | X\_GMAC\_RXD\_0 | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I | GMAC\_RXD\_0/I |
| 120 | X\_GMAC\_RXD\_1 | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I | GMAC\_RXD\_1/I |
| 121 | X\_GMAC\_RXD\_2 | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I | GMAC\_RXD\_2/I |
| 122 | X\_GMAC\_RXD\_3 | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I | GMAC\_RXD\_3/I |
| 123 | X\_GMAC\_RXD\_4 | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I | GMAC\_RXD\_4/I |
| 124 | X\_GMAC\_RXD\_5 | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I | GMAC\_RXD\_5/I |
| 125 | X\_GMAC\_RXD\_6 | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I | GMAC\_RXD\_6/I |
| 126 | X\_GMAC\_RXD\_7 | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I | GMAC\_RXD\_7/I |
| 127 | X\_GMAC\_TXEN | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O | GMAC\_TXEN/O |
| 128 | X\_GMAC\_TXER | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O | GMAC\_TXER/O |
| 129 | X\_IVA\_EXTCLK | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O | IVA\_EXTCLK/O |
| 130 | X\_IVA\_PIXCLK | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I | IVA\_PIXCLK/I |
| 131 | X\_IVA\_FRAME\_VALID | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I | IVA\_FRAME\_VALID/I |
| 132 | X\_IVA\_LINE\_VALID | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I | IVA\_LINE\_VALID /I |
| 133 | X\_IVA\_DATA\_0 | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I | IVA\_DATA\_0/I |
| 134 | X\_IVA\_DATA\_1 | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I | IVA\_DATA\_1/I |
| 135 | X\_IVA\_DATA\_2 | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I | IVA\_DATA\_2/I |
| 136 | X\_IVA\_DATA\_3 | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I | IVA\_DATA\_3/I |
| 137 | X\_IVA\_DATA\_4 | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I | IVA\_DATA\_4/I |
| 138 | X\_IVA\_DATA\_5 | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I | IVA\_DATA\_5/I |
| 139 | X\_IVA\_DATA\_6 | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I | IVA\_DATA\_6/I |
| 140 | X\_IVA\_DATA\_7 | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I | IVA\_DATA\_7/I |
| 141 | X\_IVA\_DATA\_8 | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I | IVA\_DATA\_8/I |
| 142 | X\_IVA\_DATA\_9 | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I | IVA\_DATA\_9/I |
| 143 | X\_IVA\_DATA\_10 | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I | IVA\_DATA\_10/I |
| 144 | X\_IVA\_DATA\_11 | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I | IVA\_DATA\_11/I |
| 145 | X\_PWM\_0 | paddr31m3/O | DBG\_SPI\_OUT\_0/O | DBG\_UART\_OUT\_0/O | DBG\_I2C\_OUT\_0/O | DBG\_SPINOR\_DBGSIG\_0/O | DBG\_IVA\_DBGSIG\_0/O |
| 146 | X\_PWM\_1 | jtagnsw/O | DBG\_SPI\_OUT\_1/O | DBG\_UART\_OUT\_1/O | DBG\_I2C\_OUT\_1/O | DBG\_SPINOR\_DBGSIG\_1/O | DBG\_IVA\_DBGSIG\_1/O |
| 147 | X\_PWM\_2 | jtagtop/O | DBG\_SPI\_OUT\_2/O | DBG\_UART\_OUT\_2/O | DBG\_I2C\_OUT\_2/O | DBG\_SPINOR\_DBGSIG\_2/O | DBG\_IVA\_DBGSIG\_2/O |
| 148 | X\_SPI0\_SCLK | pdbgswen/O | SPI0\_SCLK/B | DBG\_UART\_OUT\_3/O | DBG\_I2C\_OUT\_3/O | DBG\_SPINOR\_DBGSIG\_3/O | DBG\_IVA\_DBGSIG\_3/O |
| 149 | X\_SPI0\_MOSI | csysack/O | SPI0\_MOSI/B | DBG\_UART\_OUT\_4/O | DBG\_I2C\_OUT\_4/O | DBG\_SPINOR\_DBGSIG\_4/O | DBG\_IVA\_DBGSIG\_4/O |
| 150 | X\_SPI0\_MISO | event\_syncreqs/O | SPI0\_MISO/B | DBG\_UART\_OUT\_5/O | DBG\_I2C\_OUT\_5/O | DBG\_SPINOR\_DBGSIG\_5/O | DBG\_IVA\_DBGSIG\_5/O |
| 151 | X\_SPI0\_SS\_N | wen\_sram/O | SPI0\_SS\_N/B | DBG\_UART\_OUT\_6/O | DBG\_I2C\_OUT\_6/O | DBG\_SPINOR\_DBGSIG\_6/O | DBG\_IVA\_DBGSIG\_6/O |
| 152 | X\_SPI1\_SCLK | cen\_sram/O | SPI1\_SCLK/B | DBG\_UART\_OUT\_7/O | DBG\_I2C\_OUT\_7/O | DBG\_SPINOR\_DBGSIG\_7/O | DBG\_IVA\_DBGSIG\_7/O |
| 153 | X\_SPI1\_MOSI | a\_sram\_0/O | SPI1\_MOSI/B | DBG\_UART\_OUT\_8/O | DBG\_I2C\_OUT\_8/O | DBG\_SPINOR\_DBGSIG\_8/O | DBG\_IVA\_DBGSIG\_8/O |
| 154 | X\_SPI1\_MISO | a\_sram\_1/O | SPI1\_MISO/B | DBG\_UART\_OUT\_9/O | DBG\_I2C\_OUT\_9/O | DBG\_SPINOR\_DBGSIG\_9/O | DBG\_IVA\_DBGSIG\_9/O |
| 155 | X\_SPI1\_SS\_N | a\_sram\_2/O | SPI1\_SS\_N/B | DBG\_UART\_OUT\_10/O | DBG\_I2C\_OUT\_10/O | DBG\_SPINOR\_DBGSIG\_10/O | DBG\_IVA\_DBGSIG\_10/O |
| 156 | X\_I2C\_SCL | a\_sram\_3/O | DBG\_SPI\_OUT\_3/O | DBG\_UART\_OUT\_11/O | I2C\_SCL/B | DBG\_SPINOR\_DBGSIG\_11/O | DBG\_IVA\_DBGSIG\_11/O |
| 157 | X\_I2C\_SDA | a\_sram\_4/O | DBG\_SPI\_OUT\_4/O | DBG\_UART\_OUT\_12/O | I2C\_SDA/B | DBG\_SPINOR\_DBGSIG\_12/O | DBG\_IVA\_DBGSIG\_12/O |
| 158 | X\_UART\_TXD | a\_sram\_5/O | DBG\_SPI\_OUT\_5/O | UART\_TXD/O | DBG\_I2C\_OUT\_11/O | DBG\_SPINOR\_DBGSIG\_13/O | DBG\_IVA\_DBGSIG\_13/O |
| 159 | X\_UART\_RXD | a\_sram\_6/O | DBG\_SPI\_OUT\_6/O | UART\_RXD/I | DBG\_I2C\_OUT\_12/O | DBG\_SPINOR\_DBGSIG\_14/O | DBG\_IVA\_DBGSIG\_14/O |
| 160 | X\_GPIO\_PA\_0 | a\_sram\_7/O | DBG\_SPI\_OUT\_7/O | DBG\_UART\_OUT\_13/O | DBG\_I2C\_OUT\_13/O | DBG\_SPINOR\_DBGSIG\_15/O | DBG\_IVA\_DBGSIG\_15/O |
| 161 | X\_GPIO\_PA\_1 | q\_sram\_0/O | DBG\_SPI\_OUT\_8/O | DBG\_UART\_OUT\_14/O | DBG\_I2C\_OUT\_14/O | DBG\_SPINOR\_DBGSIG\_16/O | DBG\_IVA\_DBGSIG\_16/O |
| 162 | X\_GPIO\_PA\_2 | q\_sram\_1/O | DBG\_SPI\_OUT\_9/O | DBG\_UART\_OUT\_15/O | DBG\_I2C\_OUT\_15/O | DBG\_SPINOR\_DBGSIG\_17/O | DBG\_IVA\_DBGSIG\_17/O |
| 163 | X\_GPIO\_PA\_3 | q\_sram\_2/O | DBG\_SPI\_OUT\_10/O | DBG\_UART\_OUT\_16/O | DBG\_I2C\_OUT\_16/O | DBG\_SPINOR\_DBGSIG\_18/O | DBG\_IVA\_DBGSIG\_18/O |
| 164 | X\_GPIO\_PA\_4 | q\_sram\_3/O | DBG\_SPI\_OUT\_11/O | DBG\_UART\_OUT\_17/O | DBG\_I2C\_OUT\_17/O | DBG\_SPINOR\_DBGSIG\_19/O | DBG\_IVA\_DBGSIG\_19/O |
| 165 | X\_GPIO\_PA\_5 | q\_sram\_4/O | DBG\_SPI\_OUT\_12/O | DBG\_UART\_OUT\_18/O | DBG\_I2C\_OUT\_18/O | DBG\_SPINOR\_DBGSIG\_20/O | DBG\_IVA\_DBGSIG\_20/O |
| 166 | X\_GPIO\_PA\_6 | q\_sram\_5/O | DBG\_SPI\_OUT\_13/O | DBG\_UART\_OUT\_19/O | DBG\_I2C\_OUT\_19/O | DBG\_SPINOR\_DBGSIG\_21/O | DBG\_IVA\_DBGSIG\_21/O |
| 167 | X\_GPIO\_PA\_7 | q\_sram\_6/O | DBG\_SPI\_OUT\_14/O | DBG\_UART\_OUT\_20/O | DBG\_I2C\_OUT\_20/O | DBG\_SPINOR\_DBGSIG\_22/O | DBG\_IVA\_DBGSIG\_22/O |
| 168 | X\_GPIO\_PB\_0 | q\_sram\_7/O | DBG\_SPI\_OUT\_15/O | DBG\_UART\_OUT\_21/O | DBG\_I2C\_OUT\_21/O | DBG\_SPINOR\_DBGSIG\_23/O | DBG\_IVA\_DBGSIG\_23/O |
| 169 | X\_GPIO\_PB\_1 | d\_sram\_0/O | DBG\_SPI\_OUT\_16/O | DBG\_UART\_OUT\_22/O | DBG\_I2C\_OUT\_22/O | DBG\_SPINOR\_DBGSIG\_24/O | DBG\_IVA\_DBGSIG\_24/O |
| 170 | X\_GPIO\_PB\_2 | d\_sram\_1/O | DBG\_SPI\_OUT\_17/O | DBG\_UART\_OUT\_23/O | DBG\_I2C\_OUT\_23/O | DBG\_SPINOR\_DBGSIG\_25/O | DBG\_IVA\_DBGSIG\_25/O |
| 171 | X\_GPIO\_PB\_3 | d\_sram\_2/O | DBG\_SPI\_OUT\_18/O | DBG\_UART\_OUT\_24/O | DBG\_I2C\_OUT\_24/O | DBG\_SPINOR\_DBGSIG\_26/O | DBG\_IVA\_DBGSIG\_26/O |
| 172 | X\_GPIO\_PB\_4 | d\_sram\_3/O | DBG\_SPI\_OUT\_19/O | DBG\_UART\_OUT\_25/O | DBG\_I2C\_OUT\_25/O | DBG\_SPINOR\_DBGSIG\_27/O | DBG\_IVA\_DBGSIG\_27/O |
| 173 | X\_GPIO\_PB\_5 | d\_sram\_4/O | DBG\_SPI\_OUT\_20/O | DBG\_UART\_OUT\_26O/ | DBG\_I2C\_OUT\_26/O | DBG\_SPINOR\_DBGSIG\_28/O | DBG\_IVA\_DBGSIG\_28/O |
| 174 | X\_GPIO\_PB\_6 | d\_sram\_5/O | DBG\_SPI\_OUT\_21/O | DBG\_UART\_OUT\_27/O | DBG\_I2C\_OUT\_27/O | DBG\_SPINOR\_DBGSIG\_29/O | DBG\_IVA\_DBGSIG\_29/O |
| 175 | X\_GPIO\_PB\_7 | d\_sram\_6/O | DBG\_SPI\_OUT\_22/O | DBG\_UART\_OUT\_28/O | DBG\_I2C\_OUT\_28/O | DBG\_SPINOR\_DBGSIG\_30/O | DBG\_IVA\_DBGSIG\_30/O |
| 176 | X\_GPIO\_PC\_0 | d\_sram\_7/O | DBG\_SPI\_OUT\_23/O | DBG\_UART\_OUT\_29/O | DBG\_I2C\_OUT\_29/O | DBG\_SPINOR\_DBGSIG\_31/O | DBG\_IVA\_DBGSIG\_31/O |
| 177 | X\_GPIO\_PC\_1 | N.A. | DBG\_SPI\_OUT\_24/O | DBG\_UART\_OUT\_30/O | DBG\_I2C\_OUT\_30/O | DBG\_SPINOR\_DBGSIG\_32/O | DBG\_IVA\_DBGSIG\_32/O |
| 178 | X\_GPIO\_PC\_2 | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I | CS\_ntrst/I |
| 179 | X\_GPIO\_PC\_3 | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I | CS\_swclktck/I |
| 180 | X\_GPIO\_PC\_4 | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B | CS\_swditms\_swj/B |
| 181 | X\_GPIO\_PC\_5 | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I | CS\_tdi\_swj/I |
| 182 | X\_GPIO\_PC\_6 | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O | CS\_tdo\_swj/O |
| 183 | X\_GPIO\_PC\_7 | N.A. | DBG\_SPI\_OUT\_30/O | N.A. | N.A. | N.A. | N.A. |
| 184 | X\_SPI\_NOR\_SCLK | N.A. | DBG\_SPI\_OUT\_31/O | N.A. | N.A. | SPI\_NOR\_SCLK/B | N.A. |
| 185 | X\_SPI\_NOR\_MOSI | N.A. | N.A. | N.A. | N.A. | SPI\_NOR\_MOSI/B | N.A. |
| 186 | X\_SPI\_NOR\_MISO | N.A. | N.A. | N.A. | N.A. | SPI\_NOR\_MISO/B | N.A. |
| 187 | X\_SPI\_NOR\_SS\_N | N.A. | N.A. | N.A. | N.A. | SPI\_NOR\_SS\_N/B | N.A. |
| 188 | X\_SPI\_NOR\_SIO2\_IN | N.A. | N.A. | N.A. | N.A. | SPI\_NOR\_SIO2\_IN/I | N.A. |
| 189 | X\_SPI\_NOR\_SIO3\_IN | N.A. | N.A. | N.A. | N.A. | SPI\_NOR\_SIO3\_IN/I | N.A. |

* 1. Programming Configuration

Using 4 pins to switch the function mode: Normal Mode, Engineering Mode, DFT & MBIST Mode, Analog Test Mode, NAND-Tree Test Mode, FPGA Mode.

In Engineering Mode, support 4-bit registers configure the function mode: CoreSight Trace Mode, CoreSight JTAG Mode, IP Debug Mode.

In FPGA Mode, support 4-bit registers configure the function mode: FPGA Normal Mode, FPGA CoreSight Trace Mode, FPGA CoreSight JTAG Mode.

Table 56. Programming configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODE** | | **MODE\_SEL** | **reg\_mode\_sel** | **Description** |
| Normal | IP Cam | 4’b0000 | N.A | IP cam mode |
| Engineering | CoreSight Trace | 4’b0001 | 4’b0000 | CoreSight trace (power all on) mode |
| DDR\_DFI Debug | 4’b0001 | DDR\_DFI debug mode |
| DDR\_PAD Debug | 4’b0010 | DDR\_PAD debug mode |
| SDIO Debug | 4’b0011 | SDIO debug mode |
| GMAC Debug | 4’b0100 | GMAC debug mode |
| ISRAM Debug | 4’b0101 | ISRAM debug mode |
| DMA Debug | 4’b0110 | DMA debug mode |
| CoreSight Debug | 4’b0111 | CoreSight debug mode |
| SPI Debug | 4’b1000 | SPI debug mode |
| UART Debug | 4’b1001 | UART debug mode |
| I2C Debug | 4’b1010 | I2C debug mode |
| SPI-NOR Debug | 4’b1011 | SPI-NOR debug mode |
| IVA Debug | 4’b1100 | IVA debug mode |
| CoreSight JTAG | 4’b1111 | CoreSight JTAG mode |
| DFT & MBIST | TOP-SCAN | 4’b0010 | N.A | TOP-SCAN mode |
| CPU-SCAN | 4’b0011 | N.A | CPU-SCAN mode |
| TOP-MBIST | 4’b0100 | N.A | MBIST Top mode |
| CPU-MBIST | 4’b0101 | N.A | MBIST CPU mode |
| Analog | PLL Test | 4’b0110 | N.A | PLL test mode |
| ESD Test | 4’b0111 | N.A | ESD test mode |
| RTC Test | 4’b1000 | N.A | RTC test mode |
| NAND Tree | NAND Tree Test | 4’b1001 | N.A | IO open/short test mode |
| FPGA  (IVA pad power: 1.8V) | Normal | 4’b1111 | 4’b0000 | IP cam mode |
| CoreSight Trace | 4’b0001 | CoreSight trace (power all on) mode |
| CoreSight JTAG | 4’b1111 | CoreSight JTAG mode |

* 1. Output Timing
* Normal Mode (MODE\_SEL = 4’b0)

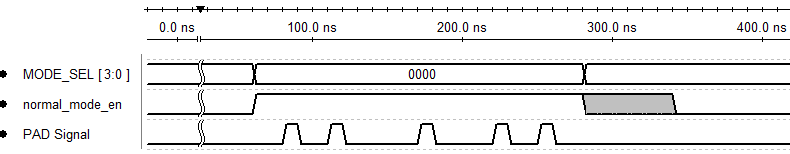


Figure 11. Normal mode output timing

* CoreSight Trace Mode (MODE\_SEL = 4’b1 & reg\_mux\_sel = 4’b0)

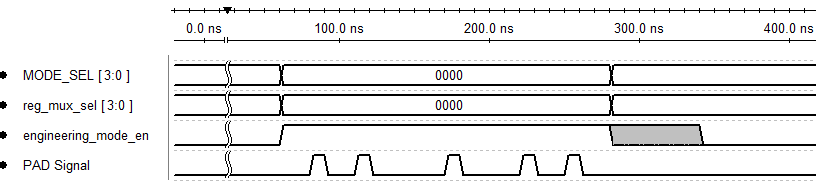


Figure 12. Engineering mode output timing

* DDR\_DFI Debug Mode (MODE\_SEL = 4’b1 & reg\_mux\_sel = 4’b1)

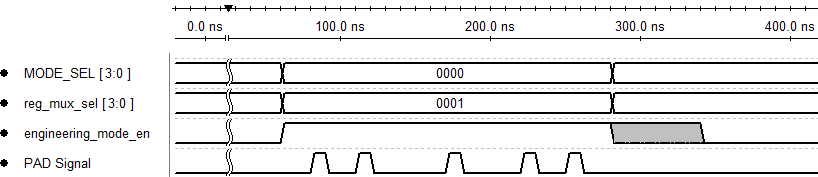


Figure 13. DDR\_DFI debug mode output timing