# VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Datasheet

## Family of Managed Gigabit Ethernet Switches

**Microsemi Headquarters**
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

©2019 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

## About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

# Contents


1 Revision History . . . 1
1.1 Revision 4.2 . . . 1
1.2 Revision 4.1 . . . 1
1.3 Revision 4.0 . . . 1
1.4 Revision 2.0 . . . 1

2 Introduction . . . 2
2.1 Register Notation . . . 2
2.2 Standard References . . . 2
2.3 Terms and Abbreviations . . . 3

3 Product Overview . . . 5
3.1 General Features . . . 5
3.1.1 Layer-2 Switching . . . 5
3.1.2 Multicast . . . 6
3.1.3 Quality of Service . . . 6
3.1.4 Security . . . 6
3.1.5 Management . . . 7
3.2 Applications . . . 7
3.3 Related Products . . . 7
3.4 Functional Overview . . . 7
3.4.1 Frame Arrival . . . 8
3.4.2 Basic and Advanced Frame Classification . . . 9
3.4.3 VCAP-II Vitesse Content Aware Processor . . . 10
3.4.4 Policing . . . 11
3.4.5 Layer-2 Forwarding . . . 11
3.4.6 Shared Queue System and Egress Scheduler . . . 12
3.4.7 Rewriter and Frame Departure . . . 13
3.4.8 CPU Port Module . . . 13
3.4.9 CPU System and Interfaces . . . 14

4 Functional Descriptions . . . 15
4.1 Port Modules . . . 15
4.1.1 Port Module Numbering and Macro Connections . . . 15
4.1.2 MAC . . . 16
4.1.3 PCS . . . 18
4.2 SERDES1G . . . 22
4.2.1 SERDES1G Basic Configuration . . . 23
4.2.2 SERDES1G Loopback Modes . . . 23
4.2.3 SERDES1G Deserializer Configuration . . . 23
4.2.4 SERDES1G Serializer Configuration . . . 24
4.2.5 SERDES1G Input Buffer Configuration . . . 25
4.2.6 SERDES1G Output Buffer Configuration . . . 25
4.2.7 SERDES1G Clock and Data Recovery (CDR) in 100BASE-FX . . . 26
4.2.8 SERDES1G Energy Efficient Ethernet . . . 26
4.2.9 SERDES1G Data Inversion . . . 26
4.3 SERDES6G . . . 26
4.3.1 SERDES6G Basic Configuration . . . 26
4.3.2 SERDES6G Loopback Modes . . . 27
4.3.3 SERDES6G Deserializer Configuration . . . 28
4.3.4 SERDES6G Serializer Configuration . . . 29

4.3.5 SERDES6G Input Buffer Configuration . . . 29
4.3.6 SERDES6G Output Buffer Configuration . . . 30
4.3.7 SERDES6G Clock and Data Recovery (CDR) in 100BASE-FX . . . 31
4.3.8 SERDES6G Energy Efficient Ethernet . . . 31
4.3.9 SERDES6G Data Inversion . . . 31
4.3.10 SERDES6G Signal Detection Enhancements . . . 31
4.3.11 SERDES6G High-Speed I/O Configuration Bus . . . 31
4.4 Copper Transceivers . . . 32
4.4.1 Register Access . . . 32
4.4.2 Cat5 Twisted Pair Media Interface . . . 33
4.4.3 LED Interface . . . 36
4.4.4 Ethernet Inline Powered Devices . . . 38
4.4.5 IEEE 802.3af PoE Support . . . 39
4.4.6 ActiPHY™ Power Management . . . 39
4.4.7 Testing Features . . . 41
4.4.8 VeriPHY™ Cable Diagnostics . . . 42
4.5 Statistics . . . 43
4.6 Classifier . . . 48
4.6.1 General Data Extraction Setup . . . 49
4.6.2 Frame Acceptance Filtering . . . 50
4.6.3 QoS, DP, and DSCP Classification . . . 51
4.6.4 VLAN Classification . . . 55
4.6.5 Link Aggregation Code Generation . . . 56
4.6.6 CPU Forwarding Determination . . . 57
4.7 VCAP-II . . . 58
4.7.1 Port Configuration . . . 61
4.7.2 VCAP IS1 . . . 62
4.7.3 VCAP IS2 . . . 67
4.7.4 VCAP ES0 . . . 75
4.7.5 Range Checkers . . . 77
4.7.6 VCAP-II Configuration . . . 78
4.7.7 Advanced VCAP Operations . . . 83
4.8 Analyzer . . . 84
4.8.1 MAC Table . . . 85
4.8.2 VLAN Table . . . 91
4.8.3 Forwarding Engine . . . 92
4.8.4 Analyzer Monitoring . . . 101
4.9 Policers and Ingress Shapers . . . 102
4.9.1 Policers . . . 102
4.9.2 Ingress Shapers . . . 104
4.10 Shared Queue System . . . 104
4.10.1 Buffer Management . . . 105
4.10.2 Frame Reference Management . . . 107
4.10.3 Resource Depletion Condition . . . 107
4.10.4 Configuration Example . . . 108
4.10.5 Watermark Programming and Consumption Monitoring . . . 108
4.10.6 Advanced Resource Management . . . 109
4.10.7 Ingress Pause Request Generation . . . 110
4.10.8 Tail Dropping . . . 111
4.10.9 Test Utilities . . . 111
4.10.10 Energy Efficient Ethernet . . . 111
4.11 Scheduler and Shaper . . . 112
4.11.1 Egress Shapers . . . 113
4.11.2 Deficit Weighted Round Robin . . . 114
4.11.3 Shaping and DWRR Scheduling Examples . . . 115
4.12 Rewriter . . . 116
4.12.1 VLAN Editing . . . 116

4.12.2 DSCP Remarking . . . 118
4.12.3 FCS Updating . . . 119
4.12.4 CPU Extraction Header Insertion . . . 119
4.13 CPU Port Module . . . 120
4.13.1 Frame Extraction . . . 121
4.13.2 Frame Injection . . . 122
4.13.3 Network Processor Interface (NPI) . . . 124
4.14 Hardware Timestamping for AVB . . . 124
4.14.1 Timestamp Classification . . . 125
4.14.2 One-Second Timer . . . 125
4.14.3 Delay Timer . . . 127
4.14.4 Time of Day Counter . . . 129
4.15 Clocking and Reset . . . 129

5 VCore-III System and CPU Interface . . . 131
5.1 VCore-III Configurations . . . 132
5.2 Clocking and Reset . . . 133
5.2.1 Watchdog Timer . . . 134
5.3 Shared Bus . . . 134
5.3.1 Shared Bus Arbitration . . . 135
5.3.2 SI Memory Region . . . 136
5.3.3 PI Memory Region . . . 137
5.3.4 DDR2 Memory Region . . . 140
5.3.5 Switch Core Registers Memory Region . . . 144
5.3.6 VCore-III Registers Memory Region . . . 144
5.4 VCore-III CPU . . . 145
5.4.1 Big Endian Support . . . 145
5.4.2 Software Debug and Development . . . 147
5.5 Manual Frame Injection and Extraction . . . 147
5.5.1 Manual Frame Extraction . . . 147
5.5.2 Manual Frame Injection . . . 149
5.5.3 Frame Interrupts . . . 150
5.6 Frame DMA . . . 150
5.6.1 DMA Control Block Structures . . . 150
5.6.2 Extraction . . . 152
5.6.3 Injection . . . 155
5.6.4 Frame DMA Interrupt . . . 158
5.7 External CPU Support . . . 159
5.7.1 Register Access and Multimaster Systems . . . 159
5.7.2 Serial Interface in Slave Mode . . . 159
5.7.3 Parallel Interface in Slave Mode . . . 161
5.7.4 MIIM Interface in Slave Mode . . . 166
5.7.5 Access to the VCore-III Shared Bus . . . 168
5.7.6 Mailbox and Semaphores . . . 169
5.8 VCore-III System Peripherals . . . 170
5.8.1 Timers . . . 170
5.8.2 UART . . . 170
5.8.3 Two-Wire Serial Interface . . . 172
5.8.4 MII Management Controller . . . 174
5.8.5 GPIO Controller . . . 176
5.8.6 Serial GPIO Controller . . . 178
5.8.7 FAN Controller . . . 182
5.8.8 Interrupt Controller . . . 184

6 Features . . . 187
6.1 Port Mapping . . . 187

6.1.1 VSC7424-02 Port Mapping . . . 187
6.1.2 VSC7425-02 Port Mapping . . . 187
6.1.3 VSC7426-02 Port Mapping . . . 188
6.1.4 VSC7427-02 Port Mapping . . . 189
6.2 Switch Control . . . 189
6.2.1 Switch Initialization . . . 189
6.3 Port Module Control . . . 189
6.3.1 MAC Configuration Port Mode Control . . . 189
6.3.2 SerDes Configuration Port Mode Control . . . 190
6.3.3 Port Reset Procedure . . . 191
6.3.4 Port Counters . . . 191
6.4 Layer-2 Switch . . . 195
6.4.1 Basic Switching . . . 195
6.4.2 Standard VLAN Operation . . . 197
6.4.3 Provider Bridges and Q-in-Q Operation . . . 200
6.4.4 Private VLANs . . . 205
6.4.5 Asymmetric VLANs . . . 209
6.4.6 Spanning Tree Protocols . . . 210
6.4.7 IEEE 802.1X: Network Access Control . . . 215
6.4.8 Link Aggregation . . . 217
6.4.9 Simple Network Management Protocol (SNMP) . . . 220
6.4.10 Mirroring . . . 220
6.5 IGMP and MLD Snooping . . . 222
6.5.1 IGMP and MLD Snooping Configuration . . . 222
6.5.2 IP Multicast Forwarding Configuration . . . 223
6.6 Quality of Service (QoS) . . . 223
6.6.1 Basic QoS Configuration . . . 224
6.6.2 IPv4 and IPv6 DSCP Remarking . . . 225
6.6.3 Voice over IP (VoIP) . . . 226
6.7 VCAP Applications . . . 227
6.7.1 Notation for Control Lists Entries . . . 227
6.7.2 Ingress Control Lists . . . 229
6.7.3 Access Control Lists . . . 229
6.7.4 Source IP Filter (SIP Filter) . . . 231
6.7.5 DHCP Application . . . 233
6.7.6 ARP Filtering . . . 234
6.7.7 Ping Policing . . . 234
6.7.8 TCP SYN Policing . . . 235
6.8 CPU Extraction and Injection . . . 235
6.8.1 Forwarding to CPU . . . 236
6.8.2 Frame Extraction . . . 237
6.8.3 Frame Injection . . . 237
6.8.4 Frame Extraction and Injection Using An External CPU . . . 238
6.9 Audio Video Bridging . . . 238
6.10 Energy Efficient Ethernet . . . 239
7 Registers . . . 241
7.1 Targets and Base Addresses . . . 241
7.2 DEVCPU_ORG . . . 242
7.2.1 DEVCPU_ORG:ORG . . . 242
7.3 SYS . . . 245
7.3.1 SYS:SYSTEM . . . 246
7.3.2 SYS:SCH . . . 253
7.3.3 SYS:SCH_LB . . . 257
7.3.4 SYS:RES_CTRL . . . 259
7.3.5 SYS:PAUSE_CFG . . . 261

7.3.6 SYS:MMGT . . . 263
7.3.7 SYS:MISC . . . 264
7.3.8 SYS:STAT . . . 264
7.3.9 SYS:PTP . . . 265
7.3.10 SYS:POL . . . 267
7.3.11 SYS:POL_MISC . . . 269
7.3.12 SYS:ISHP . . . 270
7.4 ANA . . . 272
7.4.1 ANA:ANA . . . 272
7.4.2 ANA:ANA_TABLES . . . 282
7.4.3 ANA:PORT . . . 289
7.4.4 ANA:COMMON . . . 299
7.5 REW . . . 304
7.5.1 REW:PORT . . . 304
7.5.2 REW:COMMON . . . 307
7.6 VCAP_CORE . . . 308
7.6.1 VCAP_CORE:VCAP_CORE_CFG . . . 309
7.6.2 VCAP_CORE:VCAP_CORE_CACHE . . . 313
7.6.3 VCAP_CORE:VCAP_CORE_STICKY . . . 316
7.6.4 VCAP_CORE:VCAP_CONST . . . 317
7.6.5 VCAP_CORE:TCAM_BIST . . . 319
7.7 VCAP_CORE . . . 320
7.7.1 VCAP_CORE:VCAP_CORE_CFG . . . 320
7.7.2 VCAP_CORE:VCAP_CORE_CACHE . . . 324
7.7.3 VCAP_CORE:VCAP_CORE_STICKY . . . 327
7.7.4 VCAP_CORE:VCAP_CONST . . . 328
7.7.5 VCAP_CORE:TCAM_BIST . . . 330
7.8 VCAP_CORE . . . 331
7.8.1 VCAP_CORE:VCAP_CORE_CFG . . . 331
7.8.2 VCAP_CORE:VCAP_CORE_CACHE . . . 335
7.8.3 VCAP_CORE:VCAP_CORE_STICKY . . . 338
7.8.4 VCAP_CORE:VCAP_CONST . . . 339
7.8.5 VCAP_CORE:TCAM_BIST . . . 341
7.9 DEVCPU_GCB . . . 342
7.9.1 DEVCPU_GCB:CHIP_REGS . . . 342
7.9.2 DEVCPU_GCB:SW_REGS . . . 344
7.9.3 DEVCPU_GCB:VCORE_ACCESS . . . 348
7.9.4 DEVCPU_GCB:GPIO . . . 351
7.9.5 DEVCPU_GCB:DEVCPU_RST_REGS . . . 355
7.9.6 DEVCPU_GCB:MIIM . . . 356
7.9.7 DEVCPU_GCB:MIIM_READ_SCAN . . . 361
7.9.8 DEVCPU_GCB:RAM_STAT . . . 362
7.9.9 DEVCPU_GCB:MISC . . . 362
7.9.10 DEVCPU_GCB:SIO_CTRL . . . 365
7.9.11 DEVCPU_GCB:FAN_CFG . . . 370
7.9.12 DEVCPU_GCB:FAN_STAT . . . 371
7.9.13 DEVCPU_GCB:PTP_CFG . . . 371
7.9.14 DEVCPU_GCB:PTP_STAT . . . 374
7.9.15 DEVCPU_GCB:PTP_TIMERS . . . 375
7.9.16 DEVCPU_GCB:MEMITGR . . . 377
7.10 DEVCPU_QS . . . 381
7.10.1 DEVCPU_QS:XTR . . . 381
7.10.2 DEVCPU_QS:INJ . . . 384
7.11 DEVCPU_PI . . . 388
7.11.1 DEVCPU_PI:PI . . . 388
7.12 HSIO . . . 392
7.12.1 HSIO:PLL5G_CFG . . . 393

7.12.2 HSIO:PLL5G_STATUS . . . 394
7.12.3 HSIO:RCOMP_STATUS . . . 395
7.12.4 HSIO:SERDES1G_ANA_CFG . . . 395
7.12.5 HSIO:SERDES1G_DIG_CFG . . . 401
7.12.6 HSIO:SERDES1G_DIG_STATUS . . . 402
7.12.7 HSIO:MCB_SERDES1G_CFG . . . 402
7.12.8 HSIO:SERDES6G_ANA_CFG . . . 403
7.12.9 HSIO:SERDES6G_DIG_CFG . . . 409
7.12.10 HSIO:MCB_SERDES6G_CFG . . . 411
7.13 DEV_GMII . . . 412
7.13.1 DEV_GMII:PORT_MODE . . . 412
7.13.2 DEV_GMII:MAC_CFG_STATUS . . . 413
7.14 DEV . . . 421
7.14.1 DEV:PORT_MODE . . . 421
7.14.2 DEV:MAC_CFG_STATUS . . . 423
7.14.3 DEV:PCS1G_CFG_STATUS . . . 430
7.14.4 DEV:PCS1G_TSTPAT_CFG_STATUS . . . 438
7.14.5 DEV:PCS_FX100_CONFIGURATION . . . 440
7.14.6 DEV:PCS_FX100_STATUS . . . 441
7.15 ICPU_CFG . . . 443
7.15.1 ICPU_CFG:CPU_SYSTEM_CTRL . . . 443
7.15.2 ICPU_CFG:PI_MST . . . 446
7.15.3 ICPU_CFG:SPI_MST . . . 449
7.15.4 ICPU_CFG:INTR . . . 451
7.15.5 ICPU_CFG:GPDMA . . . 484
7.15.6 ICPU_CFG:INJ_FRM_SPC . . . 489
7.15.7 ICPU_CFG:TIMERS . . . 490
7.15.8 ICPU_CFG:MEMCTRL . . . 494
7.15.9 ICPU_CFG:TWI_DELAY . . . 504
7.16 UART . . . 505
7.16.1 UART:UART . . . 505
7.17 TWI . . . 517
7.17.1 TWI:TWI . . . 517
7.18 SBA . . . 540
7.18.1 SBA:SBA . . . 540
7.19 GPDMA . . . 543
7.19.1 GPDMA:CH . . . 543
7.19.2 GPDMA:INTR . . . 555
7.19.3 GPDMA:MISC . . . 561
7.20 PHY . . . 563
7.20.1 PHY:PHY_STD . . . 563
7.20.2 PHY:PHY_EXT1 . . . 590
7.20.3 PHY:PHY_EXT2 . . . 596
7.20.4 PHY:PHY_GP . . . 598
7.20.5 PHY:PHY_EEE . . . 601

8 Electrical Specifications . . . 605
8.1 DC Characteristics . . . 605
8.1.1 Internal Pull-Up or Pull-Down Resistors . . . 605
8.1.2 Reference Clock . . . 605
8.1.3 DDR2 SDRAM Interface . . . 605
8.1.4 SGMII DC Definitions and Test Circuits . . . 606
8.1.5 Enhanced SerDes Interface . . . 607
8.1.6 SerDes (SGMII) Interface . . . 608
8.1.7 MIIM, GPIO, SI, JTAG, and Miscellaneous Signals . . . 611
8.1.8 Thermal Diode . . . 611
8.2 AC Characteristics . . . 612

8.2.1 Reference Clock . . . 612
8.2.2 Reset Timing . . . 613
8.2.3 DDR2 SDRAM Signal . . . 613
8.2.4 Enhanced SerDes Interface . . . 615
8.2.5 SerDes (SGMII) Interface . . . 619
8.2.6 MII Management . . . 621
8.2.7 Serial CPU Interface (SI) Master Mode . . . 622
8.2.8 Serial CPU Interface (SI) for Slave Mode . . . 623
8.2.9 Parallel Interface (PI) Master Mode . . . 624
8.2.10 Parallel Interface (PI) Slave Mode . . . 627
8.2.11 JTAG Interface . . . 629
8.2.12 Serial Inputs/Outputs . . . 630
8.2.13 Two-Wire Serial Interface . . . 631
8.3 Current and Power Consumption . . . 633
8.3.1 Current Consumption . . . 633
8.3.2 Power Consumption . . . 634
8.3.3 Power Supply Sequencing . . . 634
8.4 Operating Conditions . . . 635
8.5 Stress Ratings . . . 635

9 Pin Descriptions for VSC7424-02 . . . 637
9.1 Pin Diagram for VSC7424-02 . . . 637
9.2 Pins by Function for VSC7424-02 . . . 638
9.2.1 Analog Bias Signals . . . 639
9.2.2 Clock Circuits . . . 639
9.2.3 DDR2 SDRAM Interface . . . 640
9.2.4 General-Purpose Inputs and Outputs . . . 640
9.2.5 JTAG Interface . . . 642
9.2.6 MII Management Interface . . . 642
9.2.7 Miscellaneous Signals . . . 642
9.2.8 Parallel Interface . . . 643
9.2.9 Power Supplies and Ground . . . 644
9.2.10 Serial CPU Interface . . . 644
9.2.11 SerDes Interface . . . 645
9.2.12 Twisted Pair Interface . . . 646
9.3 Pins by Number for VSC7424-02 . . . 648
9.4 Pins by Name for VSC7424-02 . . . 654

10 Pin Descriptions for VSC7425-02 . . . 660
10.1 Pin Diagram for VSC7425-02 . . . 660
10.2 Pins by Function for VSC7425-02 . . . 661
10.2.1 Analog Bias Signals . . . 662
10.2.2 Clock Circuits . . . 662
10.2.3 DDR2 SDRAM Interface . . . 663
10.2.4 General-Purpose Inputs and Outputs . . . 663
10.2.5 JTAG Interface . . . 665
10.2.6 MII Management Interface . . . 665
10.2.7 Miscellaneous Signals . . . 665
10.2.8 Parallel Interface . . . 666
10.2.9 Power Supplies and Ground . . . 667
10.2.10 Serial CPU Interface . . . 667
10.2.11 SerDes Interface . . . 668
10.2.12 Enhanced SerDes Interface . . . 668
10.2.13 Twisted Pair Interface . . . 669
10.3 Pins by Number for VSC7425-02 . . . 671
10.4 Pins by Name for VSC7425-02 . . . 677

11 Pin Descriptions for VSC7426-02 . . . 683
11.1 Pin Diagram for VSC7426-02 . . . 683
11.2 Pins by Function for VSC7426-02 . . . 684
11.2.1 Analog Bias Signals . . . 685
11.2.2 Clock Circuits . . . 685
11.2.3 DDR2 SDRAM Interface . . . 686
11.2.4 General-Purpose Inputs and Outputs . . . 686
11.2.5 JTAG Interface . . . 688
11.2.6 MII Management Interface . . . 688
11.2.7 Miscellaneous Signals . . . 688
11.2.8 Parallel Interface . . . 689
11.2.9 Power Supplies and Ground . . . 690
11.2.10 Serial CPU Interface . . . 690
11.2.11 Enhanced SerDes Interface . . . 691
11.2.12 Twisted Pair Interface . . . 692
11.3 Pins by Number for VSC7426-02 . . . 694
11.4 Pins by Name for VSC7426-02 . . . 700
12 Pin Descriptions for VSC7427-02 . . . 706
12.1 Pin Diagram for VSC7427-02 . . . 706
12.2 Pins by Function for VSC7427-02 . . . 707
12.2.1 Analog Bias Signals . . . 708
12.2.2 Clock Circuits . . . 708
12.2.3 DDR2 SDRAM Interface . . . 709
12.2.4 General-Purpose Inputs and Outputs . . . 709
12.2.5 JTAG Interface . . . 711
12.2.6 MII Management Interface . . . 711
12.2.7 Miscellaneous Signals . . . 711
12.2.8 Parallel Interface . . . 712
12.2.9 Power Supplies and Ground . . . 713
12.2.10 Serial CPU Interface . . . 713
12.2.11 SerDes Interface . . . 714
12.2.12 Enhanced SerDes Interface . . . 714
12.2.13 Twisted Pair Interface . . . 715
12.3 Pins by Number for VSC7427-02 . . . 718
12.4 Pins by Name for VSC7427-02 . . . 724
13 Package Information . . . 730
13.1 Package Drawing . . . 730
13.2 Thermal Specifications . . . 731
13.3 Moisture Sensitivity . . . 732
14 Design Guidelines . . . 733
14.1 Power Supplies . . . 733
14.2 Power Supply Decoupling . . . 733
14.3 Reference Clock . . . 733
14.3.1 Single-Ended RefClk Input . . . 733
14.4 Interfaces . . . 734
14.4.1 General Recommendations . . . 734
14.4.2 SGMII Interface . . . 735
14.4.3 Parallel Interface . . . 735
14.4.4 Serial Interface . . . 735
14.4.5 Enhanced SerDes Interface . . . 735
14.4.6 Two-Wire Serial Interface . . . 736
14.4.7 DDR2 SDRAM Interface . . . 736

14.4.8 Thermal Diode External Connection . . . . . 737

15 Design Considerations . . . . . 739
15.1 10BASE-T mode unable to re-establish link . . . . . 739
15.2 Software script for link performance . . . . . 739
15.3 10BASE-T signal amplitude . . . . . 739
15.4 Clause 45 register 7.60 . . . . . 739
15.5 Clause 45 register 3.22 . . . . . 739
15.6 Clause 45 register 3.1 . . . . . 739
15.7 Clause 45 register address post-increment . . . . . 740
15.8 IEEE1588 Out of Sync Situation . . . . . 740
15.8.1 Copper Port (internal CuPHY 10-11 and External PHYs Without Timestamping) . . . . . 740
15.8.2 Serdes Port (SFP) . . . . . 741

16 Ordering Information . . . . . 742

# Figures


Figure 1 VSC7427-02 Block Diagram . . . 8
Figure 2 Basic and Advanced Frame Classification . . . 9
Figure 3 VCAP-II Security Enforcement . . . 11
Figure 4 Egress Scheduler and Shaper . . . 12
Figure 5 Advanced VLAN Tagging . . . 13
Figure 6 SERDES1G Loopback Modes . . . 23
Figure 7 SERDES Loopback . . . 28
Figure 8 Register Space Layout . . . 33
Figure 9 Cat5 Media Interface . . . 34
Figure 10 Energy Efficient Ethernet . . . 36
Figure 11 Inline Powered Ethernet Switch . . . 39
Figure 12 ActiPHY State Diagram . . . 40
Figure 13 Far-End Loopback Diagram . . . 42
Figure 14 Near-End Loopback Diagram . . . 42
Figure 15 Connector Loopback Diagram . . . 42
Figure 16 Counter Layout . . . 48
Figure 17 VLAN Acceptance Filter . . . 51
Figure 18 QoS and DP Basic Classification Flow Chart . . . 53
Figure 19 Basic DSCP Classification Flow Chart . . . 54
Figure 20 Basic VLAN Classification Flow Chart . . . 56
Figure 21 VCAP Functional Overview . . . 59
Figure 22 IS2 Entry Type Overview . . . 68
Figure 23 VCAP Configuration Overview . . . 79
Figure 24 Entry Layout In Register Example . . . 80
Figure 25 Entry Layout In Register Using Subwords Example . . . 81
Figure 26 Action Layout in Register Example . . . 82
Figure 27 Counter Layout in Register Example . . . 82
Figure 28 Move Up Operation Example . . . 83
Figure 29 MAC Table Organization . . . 86
Figure 30 Analysis Steps . . . 93
Figure 31 Frame Reference . . . 107
Figure 32 Watermark Layout . . . 109
Figure 33 Low Power Idle Operation . . . 112
Figure 34 Egress Scheduler and Shapers . . . 113
Figure 35 CPU Injection And Extraction . . . 120
Figure 36 One-Second Timer Block Diagram . . . 126
Figure 37 VCore-III System Block Diagram . . . 132
Figure 38 Shared Bus Memory Map . . . 135
Figure 39 SI Controller Memory Map . . . 136
Figure 40 SI Read Timing in Normal Mode . . . 137
Figure 41 SI Read Timing in Fast Mode . . . 137
Figure 42 PI Write Timing . . . 139
Figure 43 PI Read Timing . . . 139
Figure 44 Device-Paced PI Example . . . 140
Figure 45 16-Bit Access in Little Endian and Big Endian Modes . . . 146
Figure 46 32-Bit Access in Little Endian and Big Endian Mode . . . 146
Figure 47 General DCB Layout . . . 151
Figure 48 DCB Chain Examples . . . 152
Figure 49 Extraction DCB Layout . . . 153
Figure 50 Injection DCB Layout . . . 155
Figure 51 Write Sequence for SI . . . 160
Figure 52 Read Sequence for SI_Clk Slow . . . 161
Figure 53 Read Sequence for SI_Clk Pause . . . 161
Figure 54 Read Sequence for One-Byte Padding . . . 161

Figure 55 Write Sequence for PI . . . 163
Figure 56 Read Sequence for PI . . . 164
Figure 57 PI Read Sequence Using PI_nDone . . . 164
Figure 58 MIIM Slave Write Sequence . . . 167
Figure 59 MIIM Slave Read Sequence . . . 167
Figure 60 UART Timing . . . 171
Figure 61 Two-Wire Serial Interface Timing for 7-bit Address Access . . . 173
Figure 62 MII Management Timing . . . 175
Figure 63 SIO Timing . . . 180
Figure 64 SIO Timing with SGPIOs Disabled . . . 180
Figure 65 SIO Output Order . . . 181
Figure 66 Link Activity Timing . . . 182
Figure 67 Logical Equivalent for Interrupt Outputs . . . 186
Figure 68 Logical Equivalent for Interrupt Sources . . . 186
Figure 69 MAN Access Switch Setup . . . 203
Figure 70 ISP Example for Private VLAN . . . 207
Figure 71 DMZ Example for Private VLAN . . . 208
Figure 72 Asymmetric VLANs . . . 209
Figure 73 Spanning Tree Example . . . 211
Figure 74 Multiple Spanning Tree Example . . . 213
Figure 75 Link Aggregation Example . . . 219
Figure 76 Port Mirroring Example . . . 221
Figure 77 Resulting ACL for Lookup with PAG = (A) and IGR_PORT_MASK = (1<<8) . . . 231
Figure 78 CPU Extraction and Injection . . . 236
Figure 79 SGMII DC Input Definitions . . . 606
Figure 80 SGMII DC Transmit Test Circuit . . . 606
Figure 81 SGMII DC Definitions . . . 607
Figure 82 SGMII DC Driver Output Impedance Test Circuit . . . 607
Figure 83 nReset Signal Timing Specifications . . . 613
Figure 84 DDR2 SDRAM Input Timing Diagram . . . 614
Figure 85 DDR2 SDRAM Output Timing Diagram . . . 614
Figure 86 Test Load Circuit for DDR2 Outputs . . . 615
Figure 87 QSGMII Transient Parameters . . . 616
Figure 88 SGMII Transient Parameters . . . 619
Figure 89 MIIM Timing Diagram . . . 621
Figure 90 SI Timing Diagram for Master Mode . . . 622
Figure 91 SI Input Data Timing Diagram for Slave Mode . . . 623
Figure 92 SI Output Data Timing Diagram for Slave Mode . . . 623
Figure 93 SI_DO Disable Test Circuit . . . 624
Figure 94 VCore-III CPU External PI Read Access Timing Diagram . . . 625
Figure 95 VCore-III CPU ROM/Flash Write Timing Diagram . . . 626
Figure 96 PI Slave Write Cycle Timing Diagram . . . 627
Figure 97 PI Slave Read Cycle Timing Diagram . . . 628
Figure 98 Signal Disable Test Circuit . . . 629
Figure 99 JTAG Interface Timing Diagram . . . 629
Figure 100 Test Circuit for TDO Disable Time . . . 630
Figure 101 Serial I/O Timing Diagram . . . 630
Figure 102 Two-Wire Serial Read Timing Diagram . . . 631
Figure 103 Two-Wire Serial Write Timing Diagram . . . 631
Figure 104 Pin Diagram for VSC7424-02, Top Left . . . 637
Figure 105 Pin Diagram for VSC7424-02, Top Right . . . 638
Figure 106 Pin Diagram for VSC7425-02, Top Left . . . 660
Figure 107 Pin Diagram for VSC7425-02, Top Right . . . 661
Figure 108 Pin Diagram for VSC7426-02, Top Left . . . 683
Figure 109 Pin Diagram for VSC7426-02, Top Right . . . 684
Figure 110 Pin Diagram for VSC7427-02, Top Left . . . 706
Figure 111 Pin Diagram for VSC7427-02, Top Right . . . 707
Figure 112 Package Drawing BGA . . . 731
Figure 113 2.5 V CMOS Single-Ended RefClk Input Resistor Network . . . 734

Figure 114 3.3 V CMOS Single-Ended RefClk Input Resistor Network . . . . . 734
Figure 115 DDR2 SDRAM Point-to-Point Routing . . . . . 737
Figure 116 External Temperature Monitor Connection . . . . . 738

# Tables


Table 1 Referenced Documents . . . 3
Table 2 Terms and Abbreviations . . . 4
Table 3 Port Mapping from Switch Core Port Module to Interface Macros . . . 15
Table 4 MAC Configuration Registers . . . 16
Table 5 Frame Aging Configuration Registers . . . 18
Table 6 PCS Configuration Registers . . . 19
Table 7 Test Pattern Registers . . . 20
Table 8 Low Power Idle Registers . . . 21
Table 9 100BASE-FX Registers . . . 21
Table 10 SERDES1G Registers . . . 22
Table 11 SERDES1G Loop Bandwidth . . . 24
Table 12 SERDES6G Registers . . . 26
Table 13 PLL Configuration . . . 27
Table 14 SERDES6 Frequency Configuration Registers . . . 27
Table 15 SERDES6G Loop Bandwidth . . . 29
Table 16 De-Emphasis and Amplitude Configuration . . . 30
Table 17 Supported MDI Pair Combinations . . . 35
Table 18 LED Modes . . . 36
Table 19 Counter Registers . . . 43
Table 20 Rx Counters in the Statistics Block . . . 43
Table 21 FIFO Drop Counters in the Statistics Block . . . 45
Table 22 Tx Counters in the Statistics Block . . . 46
Table 23 General Data Extraction Registers . . . 49
Table 24 Frame Acceptance Filtering Registers . . . 50
Table 25 QoS, DP, and DSCP Classification Registers . . . 51
Table 26 VLAN Configuration Registers . . . 55
Table 27 Aggregation Code Generation Registers . . . 57
Table 28 CPU Forwarding Determination . . . 57
Table 29 Frame Type Definitions for CPU Forwarding . . . 58
Table 30 VCAP Frame Types . . . 60
Table 31 Port Module Configuration of VCAP . . . 61
Table 32 Hierarchy of IS2 Entry Types . . . 62
Table 33 IS1 Key . . . 63
Table 34 SMAC_SIP6 Key . . . 65
Table 35 SMAC_SIP4 Key . . . 65
Table 36 IS1 Action Fields . . . 65
Table 37 IS1 SMAC_SIP4 and SMAC_SIP6 Action Fields . . . 67
Table 38 IS2 Common Key Fields . . . 69
Table 39 IS2 MAC_ETYPE Key . . . 69
Table 40 IS2 MAC_LLC Key . . . 70
Table 41 IS2 MAC_SNAP Key . . . 70
Table 42 IS2 ARP Key . . . 70
Table 43 IS2 IP4_TCP_UDP Key . . . 71
Table 44 IS2 IP4_OTHER Key . . . 73
Table 45 IS2 IP6_STD Key . . . 73
Table 46 IS2 Action Fields . . . 74
Table 47 MASK_MODE and PORT_MASK Combinations . . . 75
Table 48 ES0 VID Key . . . 76
Table 49 ES0 Action Fields . . . 76
Table 50 Range Checker Configuration . . . 77
Table 51 VCAP Configuration . . . 78
Table 52 VCAP Constants . . . 78
Table 53 VCAP Parameters . . . 79
Table 54 Entry, Type, and Type-Group Parameters . . . 80

Table 55 Action and Type Field Parameters . . . 81
Table 56 Internal Mapping of Entry and Mask . . . 82
Table 57 MAC Table Access . . . 85
Table 58 MAC Table Entry . . . 86
Table 59 MAC Table Commands . . . 87
Table 60 IPv4 Multicast Destination Mask . . . 89
Table 61 IPv6 Multicast Destination Mask . . . 89
Table 62 VID/Port Filters . . . 90
Table 63 FID Definition Registers . . . 90
Table 64 Learn Limit Definition Registers . . . 91
Table 65 VLAN Table Access . . . 91
Table 66 Fields in the VLAN Table . . . 91
Table 67 VLAN Table Commands . . . 92
Table 68 DMAC Analysis Registers . . . 94
Table 69 Forwarding Decisions Based on Flood Type . . . 94
Table 70 VLAN Analysis Registers . . . 95
Table 71 Analyzer Aggregation Registers . . . 96
Table 72 VCAP IS2 Action Processing . . . 97
Table 73 SMAC Learning Registers . . . 98
Table 74 Storm Policer Registers . . . 99
Table 75 Storm Policers . . . 99
Table 76 sFlow Sampling Registers . . . 100
Table 77 Mirroring Registers . . . 100
Table 78 Analyzer Monitoring . . . 101
Table 79 Policer Control Registers . . . 102
Table 80 Ingress Shaper Control Registers . . . 104
Table 81 Reservation Watermarks . . . 105
Table 82 Sharing Watermarks . . . 106
Table 83 Watermark Configuration Example . . . 108
Table 84 Resource Management . . . 109
Table 85 Energy Efficient Ethernet Control Registers . . . 111
Table 86 Scheduler and Egress Shaper Control Registers . . . 112
Table 87 Example of Mixing DWRR and Shaping . . . 115
Table 88 Example of Strict and Work-Conserving Shaping . . . 116
Table 89 VLAN Editing Registers . . . 116
Table 90 Tagging Combinations . . . 117
Table 91 DSCP Remarking Registers . . . 118
Table 92 FCS Updating Registers . . . 119
Table 93 CPU Extraction Header Insertion Registers . . . 119
Table 94 Frame Extraction Registers . . . 121
Table 95 CPU Extraction Header . . . 121
Table 96 Frame Injection Registers . . . 122
Table 97 CPU Injection Header . . . 123
Table 98 Network Processor Interface Registers . . . 124
Table 99 One-Second Timer Registers . . . 125
Table 100 Hardware Timestamping Registers . . . 127
Table 101 Time of Day Counter Registers . . . 129
Table 102 Clocking and Reset Registers . . . 129
Table 103 VCore-III Configurations . . . 132
Table 104 Clocking and Reset Configuration Registers . . . 133
Table 105 Shared Bus Configuration Registers . . . 134
Table 106 SI Controller Configuration Registers . . . 136
Table 107 Serial Interface Pins . . . 136
Table 108 PI Controller Configuration Registers . . . 138
Table 109 Parallel Interface Pins . . . 138
Table 110 DDR2 Controller Registers . . . 140
Table 111 Selected Memory Module Variables . . . 141
Table 112 Memory Controller Timing Parameters . . . 142
Table 113 Memory Controller Mode Parameters . . . 143

Table 114 Manual Frame Extraction Registers . . . 147
Table 115 Extraction Data Special Values . . . 147
Table 116 Frame Extraction Example . . . 148
Table 117 Manual Frame Injection Registers . . . 149
Table 118 Frame Injection Example . . . 150
Table 119 DAR.Offset Field Encoding . . . 156
Table 120 Injection Frame Spacing Registers . . . 158
Table 121 SI Slave Mode Register . . . 159
Table 122 SI Slave Mode Pins . . . 159
Table 123 PI Slave Mode Registers . . . 162
Table 124 PI Slave Mode Pins . . . 162
Table 125 MIIM Slave Pins . . . 166
Table 126 MIIM Registers . . . 166
Table 127 VCore-III Shared Bus Access Registers . . . 168
Table 128 Mailbox and Semaphore Registers . . . 169
Table 129 Timer Registers . . . 170
Table 130 UART Registers . . . 171
Table 131 UART Interface Pins . . . 171
Table 132 Two-Wire Serial Interface Registers . . . 172
Table 133 Two-Wire Serial Interface Pins . . . 173
Table 134 Reserved Two-Wire Serial Interface Addresses . . . 174
Table 135 MIIM Registers . . . 174
Table 136 MIIM Management Controller Pins . . . 175
Table 137 GPIO Registers . . . 176
Table 138 GPIO Mapping . . . 177
Table 139 SIO Registers . . . 178
Table 140 SIO Controller Pins . . . 179
Table 141 Blink Modes . . . 181
Table 142 Fan Controller Registers . . . 183
Table 143 Fan Controller Pins . . . 183
Table 144 Interrupt Controller Registers . . . 184
Table 145 VSC7424-02: Mapping from Port Modules to Physical Interface Pins . . . 187
Table 146 VSC7425-02 Switch Mode 0: Mapping from Port Modules to Physical Interface Pins . . . 187
Table 147 VSC7425-02 Switch Mode 2: Mapping from Port Modules to Physical Interface Pins . . . 188
Table 148 VSC7426-02: Mapping from Port Modules to Physical Interface Pins . . . 188
Table 149 VSC7427-02: Mapping from Port Modules to Physical Interface Pins . . . 189
Table 150 MAC Configuration of Port Modes for Ports with Internal PHYs . . . 189
Table 151 MAC Configuration of Port Modes for Ports with SerDes . . . 190
Table 152 SERDES6G Configuration . . . 190
Table 153 SERDES1G Configuration . . . 191
Table 154 Mapping of RMON Counters to Port Counters . . . 192
Table 155 Mandatory Counters . . . 193
Table 156 Optional Counters . . . 193
Table 157 Recommended MAC Control Counters . . . 193
Table 158 Pause MAC Control Recommended Counters . . . 193
Table 159 Mapping of SNMP Interfaces Group Counters to Port Counters . . . 194
Table 160 Mapping of SNMP Ethernet-Like Group Counters to Port Counters . . . 194
Table 161 Port Group Identifier Table Organization . . . 195
Table 162 Port Module Registers for Standard VLAN Operation . . . 197
Table 163 Analyzer Registers for Standard VLAN Operation . . . 198
Table 164 Rewriter Registers for Standard VLAN Operation . . . 198
Table 165 Port Module Configurations for Provider Bridge VLAN Operation . . . 200
Table 166 System Configurations for Provider Bridge VLAN Operation . . . 201
Table 167 Analyzer Configurations for Provider Bridge VLAN Operation . . . 201
Table 168 Private VLAN Configuration Registers . . . 205
Table 169 Analyzer Configurations for RSTP Support . . . 210
Table 170 RSTP Port State Properties . . . 211
Table 171 RSTP Port State Configuration for Port p . . . 212
Table 172 Analyzer Configurations for MSTP Support . . . 213

Table 173 MSTP Port State Properties . . . 214
Table 174 MSTP Port State Configuration for Port p and VLAN v . . . 214
Table 175 Configurations for Port-Based Network Access Control . . . 215
Table 176 Configurations for MAC-Based Network Access Control with Secure CPU-Based Learning . . . 216
Table 177 Configurations for MAC-Based Network Access Control with No Learning . . . 217
Table 178 Link Aggregation Group Configuration Registers . . . 218
Table 179 Configuration Registers for LACP Frame Redirection to the CPU . . . 220
Table 180 System Registers for SNMP Support . . . 220
Table 181 Analyzer Registers for SNMP Support . . . 220
Table 182 Configuration Registers for Mirroring . . . 221
Table 183 Configuration Registers for IGMP and MLD Frame Redirection to CPU . . . 222
Table 184 IP Multicast Configuration Registers . . . 223
Table 185 Basic QoS Configuration Registers . . . 224
Table 186 Configuration Registers for DSCP Remarking . . . 226
Table 187 Control Lists and Application . . . 227
Table 188 Advanced QoS Configuration Register Overview . . . 229
Table 189 Configurations for Redirecting or Copying Frames to the CPU . . . 236
Table 190 Configuration Registers When Using An External CPU . . . 238
Table 191 Configuration Registers When Using Energy Efficient Ethernet . . . 239
Table 192 List of Targets and Base Addresses . . . 241
Table 193 Register Groups in DEVCPU_ORG . . . 242
Table 194 Registers in ORG . . . 242
Table 195 Fields in ERR_ACCESS_DROP . . . 243
Table 196 Fields in ERR_TGT . . . 244
Table 197 Fields in ERR_CNTS . . . 244
Table 198 Fields in CFG_STATUS . . . 245
Table 199 Register Groups in SYS . . . 245
Table 200 Registers in SYSTEM . . . 246
Table 201 Fields in RESET_CFG . . . 247
Table 202 Fields in VLAN_ETYPE_CFG . . . 247
Table 203 Fields in PORT_MODE . . . 247
Table 204 Fields in FRONT_PORT_MODE . . . 248
Table 205 Fields in SWITCH_PORT_MODE . . . 248
Table 206 Fields in FRM_AGING . . . 249
Table 207 Fields in STAT_CFG . . . 249
Table 208 Fields in EEE_CFG . . . 250
Table 209 Fields in EEE_THRES . . . 251
Table 210 Fields in IGR_NO_SHARING . . . 251
Table 211 Fields in EGR_NO_SHARING . . . 252
Table 212 Fields in SW_STATUS . . . 252
Table 213 Fields in EQ_TRUNCATE . . . 252
Table 214 Fields in EQ_PREFER_SRC . . . 253
Table 215 Fields in EXT_CPU_CFG . . . 253
Table 216 Registers in SCH . . . 253
Table 217 Fields in LB_DWRR_FRM_ADJ . . . 254
Table 218 Fields in LB_DWRR_CFG . . . 254
Table 219 Fields in SCH_DWRR_CFG . . . 255
Table 220 Fields in SCH_SHAPING_CTRL . . . 255
Table 221 Fields in SCH_LB_CTRL . . . 257
Table 222 Fields in SCH_CPU . . . 257
Table 223 Registers in SCH_LB . . . 258
Table 224 Fields in LB_THRES . . . 258
Table 225 Fields in LB_RATE . . . 259
Table 226 Registers in RES_CTRL . . . 259
Table 227 Fields in RES_CFG . . . 260
Table 228 Fields in RES_STAT . . . 260
Table 229 Registers in PAUSE_CFG . . . 261
Table 230 Fields in PAUSE_CFG . . . 261
Table 231 Fields in PAUSE_TOT_CFG . . . 262

Table 232 Fields in ATOP . . . 262
Table 233 Fields in ATOP_TOT_CFG . . . 263
Table 234 Fields in EGR_DROP_FORCE . . . 263
Table 235 Registers in MMGT . . . 263
Table 236 Fields in MMGT . . . 264
Table 237 Fields in EQ_CTRL . . . 264
Table 238 Registers in MISC . . . 264
Table 239 Fields in REPEATER . . . 264
Table 240 Registers in STAT . . . 265
Table 241 Fields in CNT . . . 265
Table 242 Registers in PTP . . . 266
Table 243 Fields in PTP_STATUS . . . 266
Table 244 Fields in PTP_DELAY . . . 267
Table 245 Fields in PTP_CFG . . . 267
Table 246 Fields in PTP_NXT . . . 267
Table 247 Registers in POL . . . 268
Table 248 Fields in POL_PIR_CFG . . . 268
Table 249 Fields in POL_MODE_CFG . . . 268
Table 250 Fields in POL_PIR_STATE . . . 269
Table 251 Registers in POL_MISC . . . 269
Table 252 Fields in POL_FLOWC . . . 270
Table 253 Fields in POL_HYST . . . 270
Table 254 Registers in ISHP . . . 270
Table 255 Fields in ISHP_CFG . . . 271
Table 256 Fields in ISHP_MODE_CFG . . . 271
Table 257 Fields in ISHP_STATE . . . 271
Table 258 Register Groups in ANA . . . 272
Table 259 Registers in ANA . . . 272
Table 260 Fields in ADVLEARN . . . 273
Table 261 Fields in VLANMASK . . . 273
Table 262 Fields in ANAGEFIL . . . 274
Table 263 Fields in ANEVENTS . . . 274
Table 264 Fields in STORMLIMIT_BURST . . . 276
Table 265 Fields in STORMLIMIT_CFG . . . 276
Table 266 Fields in ISOLATED_PORTS . . . 277
Table 267 Fields in COMMUNITY_PORTS . . . 277
Table 268 Fields in AUTOAGE . . . 278
Table 269 Fields in MACTOPTIONS . . . 278
Table 270 Fields in LEARNDISC . . . 279
Table 271 Fields in AGENCTRL . . . 279
Table 272 Fields in MIRRORPORTS . . . 280
Table 273 Fields in EMIRRORPORTS . . . 280
Table 274 Fields in FLOODING . . . 281
Table 275 Fields in FLOODING_IPMC . . . 281
Table 276 Fields in SFLOW_CFG . . . 282
Table 277 Registers in ANA_TABLES . . . 282
Table 278 Fields in ANMOVED . . . 283
Table 279 Fields in MACHDATA . . . 283
Table 280 Fields in MACLDATA . . . 283
Table 281 Fields in MACACCESS . . . 285
Table 282 Fields in MACTINDX . . . 286
Table 283 Fields in VLANACCESS . . . 286
Table 284 Fields in VLANTIDX . . . 287
Table 285 Fields in PGID . . . 288
Table 286 Fields in ENTRYLIM . . . 288
Table 287 Fields in PTP_ID_HIGH . . . 289
Table 288 Fields in PTP_ID_LOW . . . 289
Table 289 Registers in PORT . . . 289
Table 290 Fields in VLAN_CFG . . . 290

Table 291 Fields in DROP_CFG . . . 291
Table 292 Fields in QOS_CFG . . . 292
Table 293 Fields in VCAP_CFG . . . 292
Table 294 Fields in QOS_PCP_DEI_MAP_CFG . . . 295
Table 295 Fields in CPU_FWD_CFG . . . 295
Table 296 Fields in CPU_FWD_BPDU_CFG . . . 296
Table 297 Fields in CPU_FWD_GARP_CFG . . . 296
Table 298 Fields in CPU_FWD_CCM_CFG . . . 296
Table 299 Fields in PORT_CFG . . . 296
Table 300 Fields in POL_CFG . . . 298
Table 301 Registers in COMMON . . . 300
Table 302 Fields in AGGR_CFG . . . 300
Table 303 Fields in CPUQ_CFG . . . 301
Table 304 Fields in CPUQ_8021_CFG . . . 302
Table 305 Fields in DSCP_CFG . . . 302
Table 306 Fields in DSCP_REWR_CFG . . . 303
Table 307 Fields in VCAP_RNG_TYPE_CFG . . . 303
Table 308 Fields in VCAP_RNG_VAL_CFG . . . 303
Table 309 Register Groups in REW . . . 304
Table 310 Registers in PORT . . . 304
Table 311 Fields in PORT_VLAN_CFG . . . 304
Table 312 Fields in TAG_CFG . . . 305
Table 313 Fields in PORT_CFG . . . 306
Table 314 Fields in DSCP_CFG . . . 307
Table 315 Fields in PCP_DEI_QOS_MAP_CFG . . . 307
Table 316 Registers in COMMON . . . 308
Table 317 Fields in DSCP_REMAP_DP1_CFG . . . 308
Table 318 Fields in DSCP_REMAP_CFG . . . 308
Table 319 Register Groups in VCAP_CORE . . . 308
Table 320 Registers in VCAP_CORE_CFG . . . 309
Table 321 Fields in VCAP_UPDATE_CTRL . . . 311
Table 322 Fields in VCAP_MV_CFG . . . 313
Table 323 Registers in VCAP_CORE_CACHE . . . 314
Table 324 Fields in VCAP_ENTRY_DAT . . . 314
Table 325 Fields in VCAP_MASK_DAT . . . 315
Table 326 Fields in VCAP_ACTION_DAT . . . 315
Table 327 Fields in VCAP_CNT_DAT . . . 316
Table 328 Fields in VCAP_TG_DAT . . . 316
Table 329 Registers in VCAP_CORE_STICKY . . . 316
Table 330 Fields in VCAP_STICKY . . . 317
Table 331 Registers in VCAP_CONST . . . 317
Table 332 Fields in ENTRY_WIDTH . . . 317
Table 333 Fields in ENTRY_CNT . . . 317
Table 334 Fields in ENTRY_SWCNT . . . 318
Table 335 Fields in ENTRY_TG_WIDTH . . . 318
Table 336 Fields in ACTION_DEF_CNT . . . 318
Table 337 Fields in ACTION_WIDTH . . . 318
Table 338 Fields in CNT_WIDTH . . . 319
Table 339 Registers in TCAM_BIST . . . 319
Table 340 Fields in TCAM_CTRL . . . 319
Table 341 Fields in TCAM_STAT . . . 319
Table 342 Register Groups in VCAP_CORE . . . 320
Table 343 Registers in VCAP_CORE_CFG . . . 320
Table 344 Fields in VCAP_UPDATE_CTRL . . . 322
Table 345 Fields in VCAP_MV_CFG . . . 324
Table 346 Registers in VCAP_CORE_CACHE . . . 325
Table 347 Fields in VCAP_ENTRY_DAT . . . 325
Table 348 Fields in VCAP_MASK_DAT . . . 326
Table 349 Fields in VCAP_ACTION_DAT . . . 326

Table 350 Fields in VCAP_CNT_DAT . . . 327
Table 351 Fields in VCAP_TG_DAT . . . 327
Table 352 Registers in VCAP_CORE_STICKY . . . 327
Table 353 Fields in VCAP_STICKY . . . 328
Table 354 Registers in VCAP_CONST . . . 328
Table 355 Fields in ENTRY_WIDTH . . . 328
Table 356 Fields in ENTRY_CNT . . . 328
Table 357 Fields in ENTRY_SWCNT . . . 329
Table 358 Fields in ENTRY_TG_WIDTH . . . 329
Table 359 Fields in ACTION_DEF_CNT . . . 329
Table 360 Fields in ACTION_WIDTH . . . 329
Table 361 Fields in CNT_WIDTH . . . 330
Table 362 Registers in TCAM_BIST . . . 330
Table 363 Fields in TCAM_CTRL . . . 330
Table 364 Fields in TCAM_STAT . . . 330
Table 365 Register Groups in VCAP_CORE . . . 331
Table 366 Registers in VCAP_CORE_CFG . . . 331
Table 367 Fields in VCAP_UPDATE_CTRL . . . 333
Table 368 Fields in VCAP_MV_CFG . . . 335
Table 369 Registers in VCAP_CORE_CACHE . . . 336
Table 370 Fields in VCAP_ENTRY_DAT . . . 336
Table 371 Fields in VCAP_MASK_DAT . . . 337
Table 372 Fields in VCAP_ACTION_DAT . . . 337
Table 373 Fields in VCAP_CNT_DAT . . . 338
Table 374 Fields in VCAP_TG_DAT . . . 338
Table 375 Registers in VCAP_CORE_STICKY . . . 338
Table 376 Fields in VCAP_STICKY . . . 339
Table 377 Registers in VCAP_CONST . . . 339
Table 378 Fields in ENTRY_WIDTH . . . 339
Table 379 Fields in ENTRY_CNT . . . 339
Table 380 Fields in ENTRY_SWCNT . . . 340
Table 381 Fields in ENTRY_TG_WIDTH . . . 340
Table 382 Fields in ACTION_DEF_CNT . . . 340
Table 383 Fields in ACTION_WIDTH . . . 340
Table 384 Fields in CNT_WIDTH . . . 341
Table 385 Registers in TCAM_BIST . . . 341
Table 386 Fields in TCAM_CTRL . . . 341
Table 387 Fields in TCAM_STAT . . . 341
Table 388 Register Groups in DEVCPU_GCB . . . 342
Table 389 Registers in CHIP_REGS . . . 342
Table 390 Fields in GENERAL_PURPOSE . . . 343
Table 391 Fields in SI . . . 343
Table 392 Fields in CHIP_ID . . . 344
Table 393 Registers in SW_REGS . . . 344
Table 394 Fields in SEMA_INTR_ENA . . . 345
Table 395 Fields in SEMA_INTR_ENA_CLR . . . 345
Table 396 Fields in SEMA_INTR_ENA_SET . . . 345
Table 397 Fields in SEMA . . . 346
Table 398 Fields in SEMA_FREE . . . 346
Table 399 Fields in SW_INTR . . . 347
Table 400 Fields in MAILBOX . . . 347
Table 401 Fields in MAILBOX_CLR . . . 347
Table 402 Fields in MAILBOX_SET . . . 348
Table 403 Registers in VCORE_ACCESS . . . 348
Table 404 Fields in VA_CTRL . . . 348
Table 405 Fields in VA_ADDR . . . 349
Table 406 Fields in VA_DATA . . . 350
Table 407 Fields in VA_DATA_INCR . . . 351
Table 408 Fields in VA_DATA_INERT . . . 351

Table 409 Registers in GPIO . . . 351
Table 410 Fields in GPIO_OUT_SET . . . 352
Table 411 Fields in GPIO_OUT_CLR . . . 352
Table 412 Fields in GPIO_OUT . . . 353
Table 413 Fields in GPIO_IN . . . 353
Table 414 Fields in GPIO_OE . . . 353
Table 415 Fields in GPIO_INTR . . . 354
Table 416 Fields in GPIO_INTR_ENA . . . 354
Table 417 Fields in GPIO_INTR_IDENT . . . 354
Table 418 Fields in GPIO_ALT . . . 355
Table 419 Registers in DEVCPU_RST_REGS . . . 355
Table 420 Fields in SOFT_CHIP_RST . . . 356
Table 421 Fields in SOFT_DEVCPU_RST . . . 356
Table 422 Registers in MIIM . . . 357
Table 423 Fields in MII_STATUS . . . 357
Table 424 Fields in MII_CMD . . . 358
Table 425 Fields in MII_DATA . . . 359
Table 426 Fields in MII_CFG . . . 360
Table 427 Fields in MII_SCAN_0 . . . 360
Table 428 Fields in MII_SCAN_1 . . . 360
Table 429 Fields in MII_SCAN_LAST_RSLTS . . . 361
Table 430 Fields in MII_SCAN_LAST_RSLTS_VLD . . . 361
Table 431 Registers in MIIM_READ_SCAN . . . 361
Table 432 Fields in MII_SCAN_RSLTS_STICKY . . . 362
Table 433 Registers in RAM_STAT . . . 362
Table 434 Fields in RAM_INTEGRITY_ERR_STICKY . . . 362
Table 435 Registers in MISC . . . 363
Table 436 Fields in MISC_CFG . . . 363
Table 437 Fields in MISC_STAT . . . 364
Table 438 Fields in PHY_SPEED_1000_STAT . . . 364
Table 439 Fields in PHY_SPEED_100_STAT . . . 364
Table 440 Fields in PHY_SPEED_10_STAT . . . 364
Table 441 Fields in DUPLEXC_PORT_STAT . . . 365
Table 442 Registers in SIO_CTRL . . . 365
Table 443 Fields in SIO_INPUT_DATA . . . 365
Table 444 Fields in SIO_INT_POL . . . 366
Table 445 Fields in SIO_PORT_INT_ENA . . . 366
Table 446 Fields in SIO_PORT_CONFIG . . . 367
Table 447 Fields in SIO_PORT_ENABLE . . . 367
Table 448 Fields in SIO_CONFIG . . . 368
Table 449 Fields in SIO_CLOCK . . . 369
Table 450 Fields in SIO_INT_REG . . . 370
Table 451 Registers in FAN_CFG . . . 370
Table 452 Fields in FAN_CFG . . . 370
Table 453 Registers in FAN_STAT . . . 371
Table 454 Fields in FAN_CNT . . . 371
Table 455 Registers in PTP_CFG . . . 372
Table 456 Fields in PTP_MISC_CFG . . . 372
Table 457 Fields in PTP_UPPER_LIMIT_CFG . . . 372
Table 458 Fields in PTP_UPPER_LIMIT_1_TIME_ADJ_CFG . . . 373
Table 459 Fields in PTP_SYNC_INTR_ENA_CFG . . . 373
Table 460 Fields in CLK_ADJ_CFG . . . 374
Table 461 Registers in PTP_STAT . . . 374
Table 462 Fields in PTP_CURRENT_TIME_STAT . . . 375
Table 463 Fields in PTP_EVT_STAT . . . 375
Table 464 Registers in PTP_TIMERS . . . 375
Table 465 Fields in PTP_TOD_SECS . . . 376
Table 466 Fields in PTP_TOD_NANOSECS . . . 376
Table 467 Fields in PTP_DELAY . . . 376

Table 468 Fields in PTP_TIMER_CTRL . . . 377
Table 469 Registers in MEMITGR . . . 377
Table 470 Fields in MEMITGR_CTRL . . . 378
Table 471 Fields in MEMITGR_STAT . . . 379
Table 472 Fields in MEMITGR_INFO . . . 379
Table 473 Fields in MEMITGR_IDX . . . 381
Table 474 Register Groups in DEVCPU_QS . . . 381
Table 475 Registers in XTR . . . 381
Table 476 Fields in XTR_FRM_PRUNING . . . 382
Table 477 Fields in XTR_GRP_CFG . . . 382
Table 478 Fields in XTR_MAP . . . 383
Table 479 Fields in XTR_RD . . . 383
Table 480 Fields in XTR_QU_FLUSH . . . 384
Table 481 Fields in XTR_DATA_PRESENT . . . 384
Table 482 Registers in INJ . . . 385
Table 483 Fields in INJ_GRP_CFG . . . 385
Table 484 Fields in INJ_WR . . . 385
Table 485 Fields in INJ_CTRL . . . 386
Table 486 Fields in INJ_STATUS . . . 387
Table 487 Fields in INJ_ERR . . . 388
Table 488 Register Groups in DEVCPU_PI . . . 388
Table 489 Registers in PI . . . 388
Table 490 Fields in PI_CTRL . . . 389
Table 491 Fields in PI_CFG . . . 390
Table 492 Fields in PI_STAT . . . 391
Table 493 Fields in PI_MODE . . . 392
Table 494 Fields in PI_SLOW_DATA . . . 392
Table 495 Register Groups in HSIO . . . 392
Table 496 Registers in PLL5G_CFG . . . 393
Table 497 Fields in PLL5G_CFG0 . . . 393
Table 498 Registers in PLL5G_STATUS . . . 394
Table 499 Fields in PLL5G_STATUS0 . . . 394
Table 500 Registers in RCOMP_STATUS . . . 395
Table 501 Fields in RCOMP_STATUS . . . 395
Table 502 Registers in SERDES1G_ANA_CFG . . . 396
Table 503 Fields in SERDES1G_DES_CFG . . . 396
Table 504 Fields in SERDES1G_IB_CFG . . . 397
Table 505 Fields in SERDES1G_OB_CFG . . . 399
Table 506 Fields in SERDES1G_SER_CFG . . . 399
Table 507 Fields in SERDES1G_COMMON_CFG . . . 400
Table 508 Fields in SERDES1G_PLL_CFG . . . 401
Table 509 Registers in SERDES1G_DIG_CFG . . . 401
Table 510 Fields in SERDES1G_MISC_CFG . . . 401
Table 511 Registers in SERDES1G_DIG_STATUS . . . 402
Table 512 Fields in SERDES1G_DFT_STATUS . . . 402
Table 513 Registers in MCB_SERDES1G_CFG . . . 403
Table 514 Fields in MCB_SERDES1G_ADDR_CFG . . . 403
Table 515 Registers in SERDES6G_ANA_CFG . . . 403
Table 516 Fields in SERDES6G_DES_CFG . . . 404
Table 517 Fields in SERDES6G_IB_CFG . . . 406
Table 518 Fields in SERDES6G_IB_CFG1 . . . 406
Table 519 Fields in SERDES6G_OB_CFG . . . 407
Table 520 Fields in SERDES6G_OB_CFG1 . . . 407
Table 521 Fields in SERDES6G_SER_CFG . . . 408
Table 522 Fields in SERDES6G_COMMON_CFG . . . 408
Table 523 Fields in SERDES6G_PLL_CFG . . . 409
Table 524 Registers in SERDES6G_DIG_CFG . . . 409
Table 525 Fields in SERDES6G_DIG_CFG . . . 410
Table 526 Fields in SERDES6G_MISC_CFG . . . 410

Table 527 Registers in MCB_SERDES6G_CFG . . . 411
Table 528 Fields in MCB_SERDES6G_ADDR_CFG . . . 411
Table 529 Register Groups in DEV_GMII . . . 412
Table 530 Registers in PORT_MODE . . . 412
Table 531 Fields in CLOCK_CFG . . . 412
Table 532 Fields in PORT_MISC . . . 413
Table 533 Registers in MAC_CFG_STATUS . . . 413
Table 534 Fields in MAC_ENA_CFG . . . 414
Table 535 Fields in MAC_MODE_CFG . . . 414
Table 536 Fields in MAC_MAXLEN_CFG . . . 415
Table 537 Fields in MAC_TAGS_CFG . . . 415
Table 538 Fields in MAC_ADV_CHK_CFG . . . 416
Table 539 Fields in MAC_IFG_CFG . . . 416
Table 540 Fields in MAC_HDX_CFG . . . 417
Table 541 Fields in MAC_FC_CFG . . . 418
Table 542 Fields in MAC_FC_MAC_LOW_CFG . . . 419
Table 543 Fields in MAC_FC_MAC_HIGH_CFG . . . 419
Table 544 Fields in MAC_STICKY . . . 420
Table 545 Register Groups in DEV . . . 421
Table 546 Registers in PORT_MODE . . . 422
Table 547 Fields in CLOCK_CFG . . . 422
Table 548 Fields in PORT_MISC . . . 422
Table 549 Registers in MAC_CFG_STATUS . . . 423
Table 550 Fields in MAC_ENA_CFG . . . 424
Table 551 Fields in MAC_MODE_CFG . . . 424
Table 552 Fields in MAC_MAXLEN_CFG . . . 424
Table 553 Fields in MAC_TAGS_CFG . . . 425
Table 554 Fields in MAC_ADV_CHK_CFG . . . 426
Table 555 Fields in MAC_IFG_CFG . . . 426
Table 556 Fields in MAC_HDX_CFG . . . 427
Table 557 Fields in MAC_FC_CFG . . . 428
Table 558 Fields in MAC_FC_MAC_LOW_CFG . . . 428
Table 559 Fields in MAC_FC_MAC_HIGH_CFG . . . 429
Table 560 Fields in MAC_STICKY . . . 429
Table 561 Registers in PCS1G_CFG_STATUS . . . 431
Table 562 Fields in PCS1G_CFG . . . 431
Table 563 Fields in PCS1G_MODE_CFG . . . 432
Table 564 Fields in PCS1G_SD_CFG . . . 432
Table 565 Fields in PCS1G_ANEG_CFG . . . 433
Table 566 Fields in PCS1G_ANEG_NP_CFG . . . 434
Table 567 Fields in PCS1G_LB_CFG . . . 434
Table 568 Fields in PCS1G_ANEG_STATUS . . . 434
Table 569 Fields in PCS1G_ANEG_NP_STATUS . . . 435
Table 570 Fields in PCS1G_LINK_STATUS . . . 435
Table 571 Fields in PCS1G_LINK_DOWN_CNT . . . 436
Table 572 Fields in PCS1G_STICKY . . . 436
Table 573 Fields in PCS1G_LPI_CFG . . . 437
Table 574 Fields in PCS1G_LPI_WAKE_ERROR_CNT . . . 437
Table 575 Fields in PCS1G_LPI_STATUS . . . 438
Table 576 Registers in PCS1G_TSTPAT_CFG_STATUS . . . 438
Table 577 Fields in PCS1G_TSTPAT_MODE_CFG . . . 439
Table 578 Fields in PCS1G_TSTPAT_STATUS . . . 440
Table 579 Registers in PCS_FX100_CONFIGURATION . . . 440
Table 580 Fields in PCS_FX100_CFG . . . 440
Table 581 Registers in PCS_FX100_STATUS . . . 441
Table 582 Fields in PCS_FX100_STATUS . . . 442
Table 583 Register Groups in ICPU_CFG . . . 443
Table 584 Registers in CPU_SYSTEM_CTRL . . . 443
Table 585 Fields in GPR . . . 444

Table 586 Fields in RESET . . . 444
Table 587 Fields in GENERAL_CTRL . . . 445
Table 588 Fields in GENERAL_STAT . . . 446
Table 589 Registers in PI_MST . . . 447
Table 590 Fields in PI_MST_CFG . . . 447
Table 591 Fields in PI_MST_CTRL . . . 447
Table 592 Fields in PI_MST_STATUS . . . 449
Table 593 Registers in SPI_MST . . . 449
Table 594 Fields in SPI_MST_CFG . . . 449
Table 595 Fields in SW_MODE . . . 450
Table 596 Registers in INTR . . . 451
Table 597 Fields in INTR . . . 453
Table 598 Fields in INTR_ENA . . . 456
Table 599 Fields in INTR_ENA_CLR . . . 457
Table 600 Fields in INTR_ENA_SET . . . 458
Table 601 Fields in INTR_RAW . . . 460
Table 602 Fields in ICPU_IRQ0_ENA . . . 461
Table 603 Fields in ICPU_IRQ0_IDENT . . . 462
Table 604 Fields in ICPU_IRQ1_ENA . . . 463
Table 605 Fields in ICPU_IRQ1_IDENT . . . 463
Table 606 Fields in EXT_IRQ0_ENA . . . 464
Table 607 Fields in EXT_IRQ0_IDENT . . . 465
Table 608 Fields in EXT_IRQ1_ENA . . . 466
Table 609 Fields in EXT_IRQ1_IDENT . . . 466
Table 610 Fields in DEV_IDENT . . . 468
Table 611 Fields in EXT_IRQ0_INTR_CFG . . . 468
Table 612 Fields in EXT_IRQ1_INTR_CFG . . . 469
Table 613 Fields in SW0_INTR_CFG . . . 470
Table 614 Fields in SW1_INTR_CFG . . . 471
Table 615 Fields in MIIM1_INTR_CFG . . . 472
Table 616 Fields in MIIM0_INTR_CFG . . . 472
Table 617 Fields in PI_SD0_INTR_CFG . . . 473
Table 618 Fields in PI_SD1_INTR_CFG . . . 474
Table 619 Fields in UART_INTR_CFG . . . 474
Table 620 Fields in TIMER0_INTR_CFG . . . 475
Table 621 Fields in TIMER1_INTR_CFG . . . 475
Table 622 Fields in TIMER2_INTR_CFG . . . 476
Table 623 Fields in FDMA_INTR_CFG . . . 476
Table 624 Fields in TWI_INTR_CFG . . . 477
Table 625 Fields in GPIO_INTR_CFG . . . 477
Table 626 Fields in SGPIO_INTR_CFG . . . 478
Table 627 Fields in DEV_ALL_INTR_CFG . . . 479
Table 628 Fields in BLK_ANA_INTR_CFG . . . 479
Table 629 Fields in XTR_RDY0_INTR_CFG . . . 480
Table 630 Fields in XTR_RDY1_INTR_CFG . . . 481
Table 631 Fields in INJ_RDY0_INTR_CFG . . . 481
Table 632 Fields in INJ_RDY1_INTR_CFG . . . 482
Table 633 Fields in INTEGRITY_INTR_CFG . . . 483
Table 634 Fields in PTP_SYNC_INTR_CFG . . . 483
Table 635 Fields in DEV_ENA . . . 484
Table 636 Registers in GPDMA . . . 484
Table 637 Fields in FDMA_CFG . . . 485
Table 638 Fields in FDMA_CH_CFG . . . 485
Table 639 Fields in FDMA_INJ_CFG . . . 486
Table 640 Fields in FDMA_XTR_CFG . . . 487
Table 641 Fields in FDMA_XTR_STAT_LAST_DCB . . . 487
Table 642 Fields in FDMA_FRM_CNT . . . 488
Table 643 Fields in FDMA_BP_TO_INT . . . 488
Table 644 Fields in FDMA_BP_TO_DIV . . . 488

Table 645 Registers in INJ_FRM_SPC . . . 489
Table 646 Fields in INJ_FRM_SPC_TMR . . . 489
Table 647 Fields in INJ_FRM_SPC_TMR_CFG . . . 489
Table 648 Fields in INJ_FRM_SPC_LACK_CNTR . . . 490
Table 649 Fields in INJ_FRM_SPC_CFG . . . 490
Table 650 Registers in TIMERS . . . 491
Table 651 Fields in WDT . . . 491
Table 652 Fields in TIMER_TICK_DIV . . . 492
Table 653 Fields in TIMER_VALUE . . . 493
Table 654 Fields in TIMER_RELOAD_VALUE . . . 493
Table 655 Fields in TIMER_CTRL . . . 493
Table 656 Registers in MEMCTRL . . . 494
Table 657 Fields in MEMCTRL_CTRL . . . 495
Table 658 Fields in MEMCTRL_CFG . . . 496
Table 659 Fields in MEMCTRL_STAT . . . 496
Table 660 Fields in MEMCTRL_REF_PERIOD . . . 497
Table 661 Fields in MEMCTRL_TIMING0 . . . 497
Table 662 Fields in MEMCTRL_TIMING1 . . . 498
Table 663 Fields in MEMCTRL_TIMING2 . . . 499
Table 664 Fields in MEMCTRL_TIMING3 . . . 500
Table 665 Fields in MEMCTRL_MR0_VAL . . . 500
Table 666 Fields in MEMCTRL_MR1_VAL . . . 501
Table 667 Fields in MEMCTRL_MR2_VAL . . . 501
Table 668 Fields in MEMCTRL_MR3_VAL . . . 501
Table 669 Fields in MEMCTRL_TERMRES_CTRL . . . 502
Table 670 Fields in MEMCTRL_DQS_DLY . . . 502
Table 671 Fields in MEMCTRL_DQS_AUTO . . . 503
Table 672 Fields in MEMPHY_CFG . . . 503
Table 673 Fields in MEMPHY_ZCAL . . . 504
Table 674 Registers in TWI_DELAY . . . 504
Table 675 Fields in TWI_CONFIG . . . 505
Table 676 Register Groups in UART . . . 505
Table 677 Registers in UART . . . 505
Table 678 Fields in RBR_THR . . . 507
Table 679 Fields in IER . . . 508
Table 680 Fields in IIR_FCR . . . 509
Table 681 Fields in LCR . . . 510
Table 682 Fields in MCR . . . 512
Table 683 Fields in LSR . . . 513
Table 684 Fields in MSR . . . 515
Table 685 Fields in SCR . . . 516
Table 686 Fields in USR . . . 516
Table 687 Register Groups in TWI . . . 517
Table 688 Registers in TWI . . . 517
Table 689 Fields in CFG . . . 518
Table 690 Fields in TAR . . . 520
Table 691 Fields in SAR . . . 521
Table 692 Fields in DATA_CMD . . . 522
Table 693 Fields in SS_SCL_HCNT . . . 523
Table 694 Fields in SS_SCL_LCNT . . . 523
Table 695 Fields in FS_SCL_HCNT . . . 523
Table 696 Fields in FS_SCL_LCNT . . . 524
Table 697 Fields in INTR_STAT . . . 524
Table 698 Fields in INTR_MASK . . . 525
Table 699 Fields in RAW_INTR_STAT . . . 525
Table 700 Fields in RX_TL . . . 530
Table 701 Fields in TX_TL . . . 530
Table 702 Fields in CLR_INTR . . . 530
Table 703 Fields in CLR_RX_UNDER . . . 531

Table 704 Fields in CLR_RX_OVER . . . 531
Table 705 Fields in CLR_TX_OVER . . . 531
Table 706 Fields in CLR_RD_REQ . . . 531
Table 707 Fields in CLR_TX_ABRT . . . 532
Table 708 Fields in CLR_RX_DONE . . . 532
Table 709 Fields in CLR_ACTIVITY . . . 532
Table 710 Fields in CLR_STOP_DET . . . 533
Table 711 Fields in CLR_START_DET . . . 533
Table 712 Fields in CLR_GEN_CALL . . . 533
Table 713 Fields in CTRL . . . 534
Table 714 Fields in STAT . . . 534
Table 715 Fields in TXFLR . . . 536
Table 716 Fields in RXFLR . . . 536
Table 717 Fields in TX_ABRT_SOURCE . . . 536
Table 718 Fields in SDA_SETUP . . . 538
Table 719 Fields in ACK_GEN_CALL . . . 539
Table 720 Fields in ENABLE_STATUS . . . 539
Table 721 Register Groups in SBA . . . 540
Table 722 Registers in SBA . . . 540
Table 723 Fields in PL1 . . . 541
Table 724 Fields in PL2 . . . 541
Table 725 Fields in PL3 . . . 541
Table 726 Fields in WT_EN . . . 542
Table 727 Fields in WT_TCL . . . 542
Table 728 Fields in WT_CL1 . . . 542
Table 729 Fields in WT_CL2 . . . 542
Table 730 Fields in WT_CL3 . . . 543
Table 731 Register Groups in GPDMA . . . 543
Table 732 Registers in CH . . . 543
Table 733 Fields in SAR . . . 544
Table 734 Fields in DAR . . . 545
Table 735 Fields in LLP . . . 545
Table 736 Fields in CTL0 . . . 546
Table 737 Fields in CTL1 . . . 549
Table 738 Fields in SSTAT . . . 549
Table 739 Fields in DSTAT . . . 550
Table 740 Fields in SSTATAR . . . 550
Table 741 Fields in DSTATAR . . . 550
Table 742 Fields in CFG0 . . . 551
Table 743 Fields in CFG1 . . . 553
Table 744 Registers in INTR . . . 555
Table 745 Fields in RAW_TFR . . . 556
Table 746 Fields in RAW_BLOCK . . . 556
Table 747 Fields in RAW_ERR . . . 557
Table 748 Fields in STATUS_TFR . . . 557
Table 749 Fields in STATUS_BLOCK . . . 557
Table 750 Fields in STATUS_ERR . . . 558
Table 751 Fields in MASK_TFR . . . 558
Table 752 Fields in MASK_BLOCK . . . 559
Table 753 Fields in MASK_ERR . . . 559
Table 754 Fields in CLEAR_TFR . . . 560
Table 755 Fields in CLEAR_BLOCK . . . 560
Table 756 Fields in CLEAR_ERR . . . 561
Table 757 Fields in STATUSINT . . . 561
Table 758 Registers in MISC . . . 562
Table 759 Fields in DMA_CFG_REG . . . 562
Table 760 Fields in CH_EN_REG . . . 562
Table 761 Fields in DMA_COMP_VERSION . . . 563
Table 762 Register Groups in PHY . . . 563

Table 763 Registers in PHY_STD . . . 563
Table 764 Fields in PHY_CTRL . . . 565
Table 765 Fields in PHY_STAT . . . 566
Table 766 Fields in PHY_IDF1 . . . 567
Table 767 Fields in PHY_IDF2 . . . 567
Table 768 Fields in PHY_AUTONEG_ADVERTISMENT . . . 567
Table 769 Fields in PHY_AUTONEG_LP_ABILITY . . . 568
Table 770 Fields in PHY_AUTONEG_EXP . . . 569
Table 771 Fields in PHY_AUTONEG_NEXTPAGE_TX . . . 569
Table 772 Fields in PHY_AUTONEG_LP_NEXTPAGE_RX . . . 570
Table 773 Fields in PHY_CTRL_1000BT . . . 570
Table 774 Fields in PHY_STAT_1000BT . . . 571
Table 775 Fields in MMD_ACCESS_CFG . . . 572
Table 776 Fields in MMD_ADDR_DATA . . . 572
Table 777 Fields in PHY_STAT_1000BT_EXT1 . . . 573
Table 778 Fields in PHY_STAT_100BTX . . . 573
Table 779 Fields in PHY_STAT_1000BT_EXT2 . . . 574
Table 780 Fields in PHY_BYPASS_CTRL . . . 575
Table 781 Fields in PHY_ERROR_CNT1 . . . 576
Table 782 Fields in PHY_ERROR_CNT2 . . . 577
Table 783 Fields in PHY_ERROR_CNT3 . . . 577
Table 784 Fields in PHY_CTRL_STAT_EXT . . . 577
Table 785 Fields in PHY_CTRL_EXT1 . . . 579
Table 786 Fields in PHY_CTRL_EXT2 . . . 580
Table 787 Fields in PHY_INT_MASK . . . 581
Table 788 Fields in PHY_INT_STAT . . . 583
Table 789 Fields in PHY_AUX_CTRL_STAT . . . 585
Table 790 Fields in PHY_LED_MODE_SEL . . . 588
Table 791 Fields in PHY_LED_BEHAVIOR_CTRL . . . 589
Table 792 Fields in PHY_MEMORY_PAGE_ACCESS . . . 590
Table 793 Registers in PHY_EXT1 . . . 590
Table 794 Fields in PHY_CRC_GOOD_CNT . . . 591
Table 795 Fields in PHY_EXT_MODE_CTRL . . . 591
Table 796 Fields in PHY_CTRL_EXT3 . . . 592
Table 797 Fields in PHY_CTRL_EXT4 . . . 594
Table 798 Fields in PHY_1000BT_EPG1 . . . 594
Table 799 Fields in PHY_1000BT_EPG2 . . . 596
Table 800 Registers in PHY_EXT2 . . . 597
Table 801 Fields in PHY_PMD_TX_CTRL . . . 597
Table 802 Fields in PHY_EEE_CTRL . . . 597
Table 803 Registers in PHY_GP . . . 598
Table 804 Fields in PHY_COMA_MODE_CTRL . . . 599
Table 805 Fields in PHY_ENHANCED_LED_CTRL . . . 599
Table 806 Fields in PHY_GLOBAL_INT_STAT . . . 600
Table 807 Registers in PHY_EEE . . . 601
Table 808 Fields in PHY_PCS_STATUS1 . . . 602
Table 809 Fields in PHY_EEE_CAPABILITIES . . . 602
Table 810 Fields in PHY_EEE_WAKE_ERR_CNT . . . 603
Table 811 Fields in PHY_EEE_ADVERTISEMENT . . . 603
Table 812 Fields in PHY_EEE_LP_ADVERTISEMENT . . . 603
Table 813 Internal Pull-Up or Pull-Down Resistors . . . 605
Table 814 Reference Clock Input DC Specifications . . . 605
Table 815 DDR2 SDRAM Signal DC Specifications . . . 606
Table 816 Enhanced SerDes Driver DC Specifications . . . 607
Table 817 Enhanced SerDes Receiver DC Specifications . . . 608
Table 818 SerDes Driver DC Specifications . . . 609
Table 819 SerDes Receiver DC Specifications . . . 610
Table 820 . . . 611
Table 821 MIIM, GPIO, SI, JTAG, and Miscellaneous Signals DC Specifications . . . 611

Table 822 Thermal Diode Parameters . . . 611
Table 823 Reference Clock AC Specifications . . . 612
Table 824 nReset Timing Specifications . . . 613
Table 825 DDR2 SDRAM Input Signal AC Characteristics . . . 614
Table 826 DDR2 SDRAM Output Signal AC Characteristics . . . 614
Table 827 Enhanced SerDes Output AC Specifications in SGMII Mode . . . 616
Table 828 Enhanced SerDes Output AC Specifications in QSGMII Mode . . . 616
Table 829 Enhanced SerDes Driver Jitter Characteristics in SGMII Mode . . . 617
Table 830 Enhanced SerDes Driver Jitter Characteristics in QSGMII Mode . . . 617
Table 831 Enhanced SerDes Input AC Specifications in SGMII Mode . . . 617
Table 832 Enhanced SerDes Input AC Specifications in QSGMII Mode . . . 617
Table 833 Enhanced SerDes Receiver Jitter Tolerance in SGMII Mode . . . 618
Table 834 Enhanced SerDes Receiver Jitter Tolerance in QSGMII Mode . . . 618
Table 835 SerDes Output AC Specifications . . . 619
Table 836 SerDes Driver Jitter Characteristics . . . 620
Table 837 SerDes Input AC Specifications . . . 620
Table 838 SerDes Receiver Jitter Tolerance . . . 620
Table 839 MIIM Timing Specifications . . . 621
Table 840 SI Timing Specifications for Master Mode . . . 622
Table 841 SI Timing Specifications for Slave Mode . . . 623
Table 842 VCore-III CPU External PI Read Timing Specifications . . . 625
Table 843 VCore-III CPU External PI Write Timing Specifications . . . 626
Table 844 PI Slave Mode Timing Specifications . . . 628
Table 845 JTAG Interface AC Specifications . . . 629
Table 846 Serial I/O Timing Specifications . . . 631
Table 847 Two-Wire Serial Interface AC Specifications . . . 632
Table 848 Operating Current for VSC7424-02 . . . 633
Table 849 Operating Current for VSC7425-02, VSC7426-02, and VSC7427-02 . . . 633
Table 850 Power Consumption for VSC7424-02 . . . 634
Table 851 Power Consumption for VSC7425-02, VSC7426-02, and VSC7427-02 . . . 634
Table 852 Recommended Operating Conditions . . . 635
Table 853 Stress Ratings . . . 635
Table 854 Pin Type Symbol Definitions . . . 638
Table 855 Analog Bias Pins . . . 639
Table 856 System Clock Interface Pins . . . 639
Table 857 DDR2 SDRAM Pins . . . 640
Table 858 GPIO Pin Mapping . . . 641
Table 859 JTAG Interface Pins . . . 642
Table 860 MII Management Interface Pins . . . 642
Table 861 Miscellaneous Pins . . . 642
Table 862 Parallel Interface VCore-III Master Mode Pins . . . 643
Table 863 Parallel CPU Interface Slave Mode Pins . . . 644
Table 864 Power Supply and Ground Pins . . . 644
Table 865 Serial CPU Interface Pins . . . 645
Table 866 SerDes Interface Pins . . . 645
Table 867 Twisted Pair Interface Pins . . . 646
Table 868 Pin Type Symbol Definitions . . . 661
Table 869 Analog Bias Pins . . . 662
Table 870 System Clock Interface Pins . . . 662
Table 871 DDR2 SDRAM Pins . . . 663
Table 872 GPIO Pin Mapping . . . 664
Table 873 JTAG Interface Pins . . . 665
Table 874 MII Management Interface Pins . . . 665
Table 875 Miscellaneous Pins . . . 665
Table 876 Parallel Interface VCore-III Master Mode Pins . . . 666
Table 877 Parallel CPU Interface Slave Mode Pins . . . 667
Table 878 Power Supply and Ground Pins . . . 667
Table 879 Serial CPU Interface Pins . . . 668
Table 880 SerDes Interface Pins . . . 668

Table 881 Enhanced SerDes Interface Pins . . . 668
Table 882 Twisted Pair Interface Pins . . . 669
Table 883 Pin Type Symbol Definitions . . . 684
Table 884 Analog Bias Pins . . . 685
Table 885 System Clock Interface Pins . . . 685
Table 886 DDR2 SDRAM Pins . . . 686
Table 887 GPIO Pin Mapping . . . 687
Table 888 JTAG Interface Pins . . . 688
Table 889 MII Management Interface Pins . . . 688
Table 890 Miscellaneous Pins . . . 688
Table 891 Parallel Interface VCore-III Master Mode Pins . . . 689
Table 892 Parallel CPU Interface Slave Mode Pins . . . 690
Table 893 Power Supply and Ground Pins . . . 690
Table 894 Serial CPU Interface Pins . . . 691
Table 895 Enhanced SerDes Interface Pins . . . 691
Table 896 Twisted Pair Interface Pins . . . 692
Table 897 Pin Type Symbol Definitions . . . 707
Table 898 Analog Bias Pins . . . 708
Table 899 System Clock Interface Pins . . . 708
Table 900 DDR2 SDRAM Pins . . . 709
Table 901 GPIO Pin Mapping . . . 710
Table 902 JTAG Interface Pins . . . 711
Table 903 MII Management Interface Pins . . . 711
Table 904 Miscellaneous Pins . . . 711
Table 905 Parallel Interface VCore-III Master Mode Pins . . . 712
Table 906 Parallel CPU Interface Slave Mode Pins . . . 713
Table 907 Power Supply and Ground Pins . . . 713
Table 908 Serial CPU Interface Pins . . . 714
Table 909 SerDes Interface Pins . . . 714
Table 910 Enhanced SerDes Interface Pins . . . 714
Table 911 Twisted Pair Interface Pins . . . 715
Table 912 Thermal Resistances BGA . . . 732
Table 913 Enhanced SerDes Interface Coupling Requirements . . . 736
Table 914 Recommended Skew Budget . . . 737
Table 915 Ordering Information: TQFP Package . . . 742
Table 916 Ordering Information: BGA Package . . . 742

# 1 Revision History

This section describes the changes that were implemented in this document. The changes are listed by revision, starting with the most current publication.

## 1.1 Revision 4.2

Revision 4.2 of this datasheet was published in April 2019. The following is a summary of the changes implemented in the datasheet:

- Frame Arrival section was updated. For more information, see Frame Arrival, page 8.
- MIIM Interface in Slave Mode section was updated with a note. For more information, see MIIM Interface in Slave Mode, page 166.
- April 2019VeriPHY™ Cable Diagnostics section was updated. For more information, see VeriPHY™ Cable Diagnostics, page 42.
- VeriPHY control registers were deleted. For more information, see PHY:PHY_EXT1, page 590.
- Design considerations were added to address issues with 1588 out-of-sync and copper ports. For more information, see Design Considerations, page 739.

## 1.2 Revision 4.1

Revision 4.1 of this datasheet was published in September 2014. In revision 4.1 of the document, the package drawing was updated to reflect two top views, which represent one of two packages customers can expect to receive. The maximum package height was changed to 2.44 mm. For more information, see Package Drawing, page 839.

## 1.3 Revision 4.0

Revision 4.0 of this datasheet was published in October 2012. In revision 4.0 of the document, errata items, which were previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as open issues, are now reconciled in the datasheet. Now that the information is available in the datasheet, the previously published errata document no longer applies, and it has been removed from the Vitesse Web site. For more information about the design considerations, see Design Considerations, page 739.

## 1.4 Revision 2.0

Revision 2.0 of this datasheet was published in September 2012. This was the first publication of the document.

# 2 Introduction

This document consists of descriptions and specifications for both functional and physical aspects of the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices. It is intended for system designers and software developers.

In addition to the datasheet, Microsemi maintains an extensive part-specific library of support and collateral materials that you may find useful in developing your own product. Depending upon the Microsemi device, this library may include:

- Application notes that provide detailed descriptions of the use of the particular Microsemi product to solve real-world problems
- White papers published by industry experts that provide ancillary and background information useful in developing products that take full advantage of Microsemi product designs and capabilities
- User guides that describe specific techniques for interfacing to the particular Microsemi products
- Reference designs showing the Microsemi device built in to applications in ways intended to exploit its relative strengths
- Software Development Kits with sample commands and scripts
- Presentations highlighting the operational features and specifications of the devices to assist in developing your own product road map
- Input/Output Buffer Information specification (IBIS) models to help you create and support the interfaces available on the particular Microsemi product

Visit and register as a user on the Microsemi Web site to keep abreast of the latest innovations from research and development teams and the most current product and application documentation. The address of the Microsemi Web site is www.Microsemi.com.

## 2.1 Register Notation

This datasheet uses the following general register notation:

<TARGET>:<REGISTER_GROUP>:<REGISTER>.<FIELD>

<REGISTER_GROUP> is not always present. In that case, the following notation is used:

<TARGET>::<REGISTER>.<FIELD>

When a register group does exist, it is always prepended with a target in the notation.

In sections where only one register is discussed, or the target (and register group) is known from the context, the <TARGET>:<REGISTER_GROUP>: may be omitted for brevity, and uses the following notation:

<REGISTER>.<FIELD>

Also, when a register contains only one field, the .<FIELD> is not included in the notation.

## 2.2 Standard References

This document uses the following industry references.

*Table 1 •* **Referenced Documents**

| Document | Title | Revision |
|---|---|---|
| **IEEE** | | |
| IEEE 802.1ad | 802.1Q Amendment 4: Provider Bridges | -2005 |
| IEEE 802.1D | Media Access Control (MAC) Bridges | -2004 |
| IEEE 802.1Q | Virtual Bridged Local Area Networks | -2005 |
| IEEE 802.3 | Local and metropolitan area networks — Specific requirements Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications | -2008 |
| IEEE 802.3az | Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Amendment: Media Access Control Parameters, Physical Layers and Management Parameters for Energy-Efficient Ethernet | -2010 |
| **IETF** | | |
| RFC-2236 | Internet Group Management Protocol, Version 2 (IGMPv2) | November 1997 |
| RFC-2710 | Multicast Listener Discovery for IPv6 (MLDv1) | October 1999 |
| RFC-2819 | Remote Network Monitoring (RMON) MIB | May 2000 |
| RFC-2863 | The Interfaces Group MIB | June 2000 |
| RFC-3376 | Internet Group Management Protocol, Version 3 (IGMPv3) | October 2002 |
| RFC-3635 | Definitions of Managed Objects for Ethernet-like Interface Types | September 2003 |
| **Other** | | |
| ENG-46158 | Cisco Serial GMII (SGMII) Specification | 1.7 |
| EDCS-540123 | Cisco QSGMII Specification | 1.3 |
| JESD79 | DDR2 SDRAM Specification | 2B |

## 2.3 Terms and Abbreviations

The following terms and abbreviations are used throughout this document.

***Table 2 •*** **Terms and Abbreviations**

| Term | Explanation |
|---|---|
| ACL | Access Control List |
| DEI | IEEE Drop Eligible Indicator. |
| PAG | Policy association group. Used to map many services to a shared security Policy. |
| PB | IEEE 802.1AD Provider Bridging (also known as “Q-in-Q”). |
| PCP | IEEE Priority Code Point interpretation of Ethernet Priority (also known as 802.1p) bits. |
| VCAP-II | Vitesse Content Aware Processor, TCAM-based classification and security. |
| VID | IEEE VLAN Identifier. |
| Classified VLAN | The final VLAN ID classification of a frame used in the forwarding process. The classified VLAN is the result of basic and advanced classification. |
| Basic VLAN | The VLAN ID returned by the basic classification. A basic VLAN is assigned to every frame as a default classified VLAN if no more advanced VLAN classification is carried out on the frame. |

# 3 Product Overview

The managed SparX-III family of Gigabit Ethernet switches are pin-compatible devices with port counts ranging from 10 Gigabit Ethernet ports to 26 Gigabit Ethernet ports. The switches integrate up to 12 Gigabit copper PHYs and provide both SGMII and QSGMII interfaces.

These devices provide a rich set of SME Ethernet switching features such as Layer-2 forwarding with advanced TCAM-based VLAN and QoS processing enabling delivery of differentiated services. Security is assured through frame processing using a TCAM-based Vitesse Content Aware Processor (VCAP-II). In addition, the managed SparX-III family contains a powerful 416 MHz CPU enabling full management of the switch.

The managed SparX-III family contains the following four devices:

- VSC7424-02 supports 10× 1G with 8x 1G copper PHYs + 2× 1G SGMII
- VSC7425-02 supports two major port configurations:

  18× 1G with 12× 1G copper PHYs + 2× 1G SGMII + 1× QSGMII

  18× 1G with 12× 1G copper PHYs + 6× 1G SGMII
- VSC7426-02 supports 24× 1G with 12× 1G copper PHYs + 3× QSGMII
- VSC7427-02 supports 26× 1G with 12× 1G copper PHYs + 3× QSGMII + 2× 1G SGMII

## 3.1 General Features

- All 1G Ethernet ports are tri-speed 10/100/1000 Mbps ports
- Integrated copper transceivers are compliant with IEEE 802.3ab and support Microsemi ActiPHY™ link down power savings and PerfectReach™ smart cable reach algorithm
- SGMII ports support both 100-BASE-FX and 1000-BASE-X-SERDES
- Four megabits of integrated shared packet memory
- Fully nonblocking wire-speed switching performance for all frame sizes
- Eight priorities and eight queues per port
- Policing per queue and per port
- DWRR scheduler/shaper per queue and per port with a mix of strict and weighted queues
- 256 TCAM-based egress tagging entries
- Up to 256 TCAM-based classification entries for Quality of Service (QoS) and VLAN membership
- Up to 512 host identity entries for source IP guarding
- 256 TCAM-based security enforcement entries
- Energy Efficient Ethernet (IEEE 802.3az) is supported by both the switch core and the internal copper PHYs
- Audio/Video bridging (AVB) with support for time-synchronized, low-latency audio and video streaming services
- VCore-III CPU system with integrated 416 MHz MIPS 24KEc™ CPU with MMU and DDR2 SDRAM controller

### 3.1.1 Layer-2 Switching

- 8,192 MAC addresses
- 4,096 VLANs (IEEE 802.1Q)
- Push and pop of VLAN tags
- TCAM-based VLAN classification and translation with pattern matching against Layer 2 through Layer 4 information such as MAC addresses, VLAN tag header, EtherType, DSCP, IP addresses, and TCP/UDP ports and ranges
- Up to 256 QoS and VLAN TCAM entries
- 256 VLAN egress tagging TCAM entries
- Link aggregation (IEEE 802.3ad)
- Link aggregation traffic distribution is programmable and based on Layer 2 through Layer 4 information
- Wire-speed hardware-based learning and CPU-based learning configurable per port

- Independent and shared VLAN learning
- Provider Bridging (VLAN Q-in-Q) support (IEEE 802.1ad)
- Rapid Spanning Tree Protocol support (IEEE 802.1w)
- Multiple Spanning Tree Protocol support (IEEE 802.1s)
- Jumbo frame support up to 9.6 kilobytes with programmable MTU per port

### 3.1.2 Multicast

- 8K L2 multicast group addresses with 64 port masks
- 8K IPv4/IPv6 multicast groups
- Internet Group Management Protocol version 2 (IGMPv2) support
- Internet Group Management Protocol version 3 (IGMPv3) support with source specific multicast forwarding
- Multicast Listener Discovery (MLDv1) support
- Multicast Listener Discovery (MLDv2) support with source specific forwarding (32-bit LSB of SIP used for indexing source IP address)

### 3.1.3 Quality of Service

- Eight QoS queues per port with strict or deficit weighted round-robin scheduling (DWRR)
- TCAM-based QoS classification with pattern matching against Layer 2 through Layer 4 information
- 256 QoS and VLAN TCAM entries
- DSCP translation, both ingress and/or egress
- DSCP remarking based on QoS class
- VLAN (PCP, DEI, and VID) translation, both ingress and egress
- PCP and DEI remarking based on QoS class
- Per-queue and per-port policing and shaping, programmable in steps of 100 kbps
- Per-flow policing through TCAM-based pattern matching, up to 256 policers
- Full-duplex flow control (IEEE 802.3X) and half-duplex backpressure, symmetric and asymmetric

### 3.1.4 Security

- Vitesse Content Aware Processor (VCAP-II) packet filtering engine using ACLs for ingress and egress packet inspection:

  256 security VCAP entries

  Up to 256 shared VCAP rate policers with rate measurements in frames per second or bits per second

  Eight shared range checkers supporting ranges based on TCP/UDP port numbers, DSCP values, and VLAN identifiers

  VCAP match patterns supporting generic MAC, ARP, IPv4, and IPv6 protocols

  VCAP actions including permit/deny, police, count, CPU-copy, and mirror

  Special support for IP fragments, UDP/TCP port ranges, and ARP sanity check

  Extensive CPU DoS prevention by VCAP rate policers and hit-me-once functions

  Surveillance functions supported by 32-bit VCAP counters
- Generic storm controllers for flooded broadcast, flooded multicast, and flooded unicast traffic
- Selectable CPU queues for segregation of CPU redirected traffic, with 8 queues supported
- Per-port, per-address registration for snooping of reserved IEEE MAC addresses (BPDU, GARP)
- Port-based and MAC-based access control (IEEE 802.1X)
- Per-port CPU-based learning with option for secure CPU-based learning
- Per-port ingress and egress mirroring
- Mirroring per VLAN and per VCAP match

### 3.1.5 Management

- MIPS 24KEc™ CPU system with memory management unit (MMU), and 32 kilobytes of instruction cache (I-cache) and 32 kilobytes of data cache (D-cache)
- CPU frame extraction (eight queues) and injection (two queues) through DMA, which enables efficient data transfer between Ethernet ports and CPU
- EJTAG debug interface
- Eight-bit DDR2 SDRAM interface
- Thirty-two pin-shared general-purpose I/Os
- Eight-bit parallel slave interface through GPIOs
- Serial LED controller controlling up to 32 ports with four LEDs each
- Serial GPIO controller
- PHY management controller
- Per-port 32-bit counter set with support for the RMON statistics group (RFC 2819) and SNMP interfaces group (RFC 2863)

## 3.2 Applications

VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 target the lightly and fully managed Ethernet switch equipment in the SME and Enterprise.

## 3.3 Related Products

VSC7420-02 SparX-III Gigabit Ethernet switch: 10 ports with 8 integrated PHYs and 2 SGMII

VSC7421-02 SparX-III Gigabit Ethernet switch: 17 ports with 12 integrated PHYs and 1 QSGMII

VSC7422-02 SparX-III Gigabit Ethernet Switch: 25 ports with 12 integrated PHYs, 3 QSGMII, and 1 SGMII

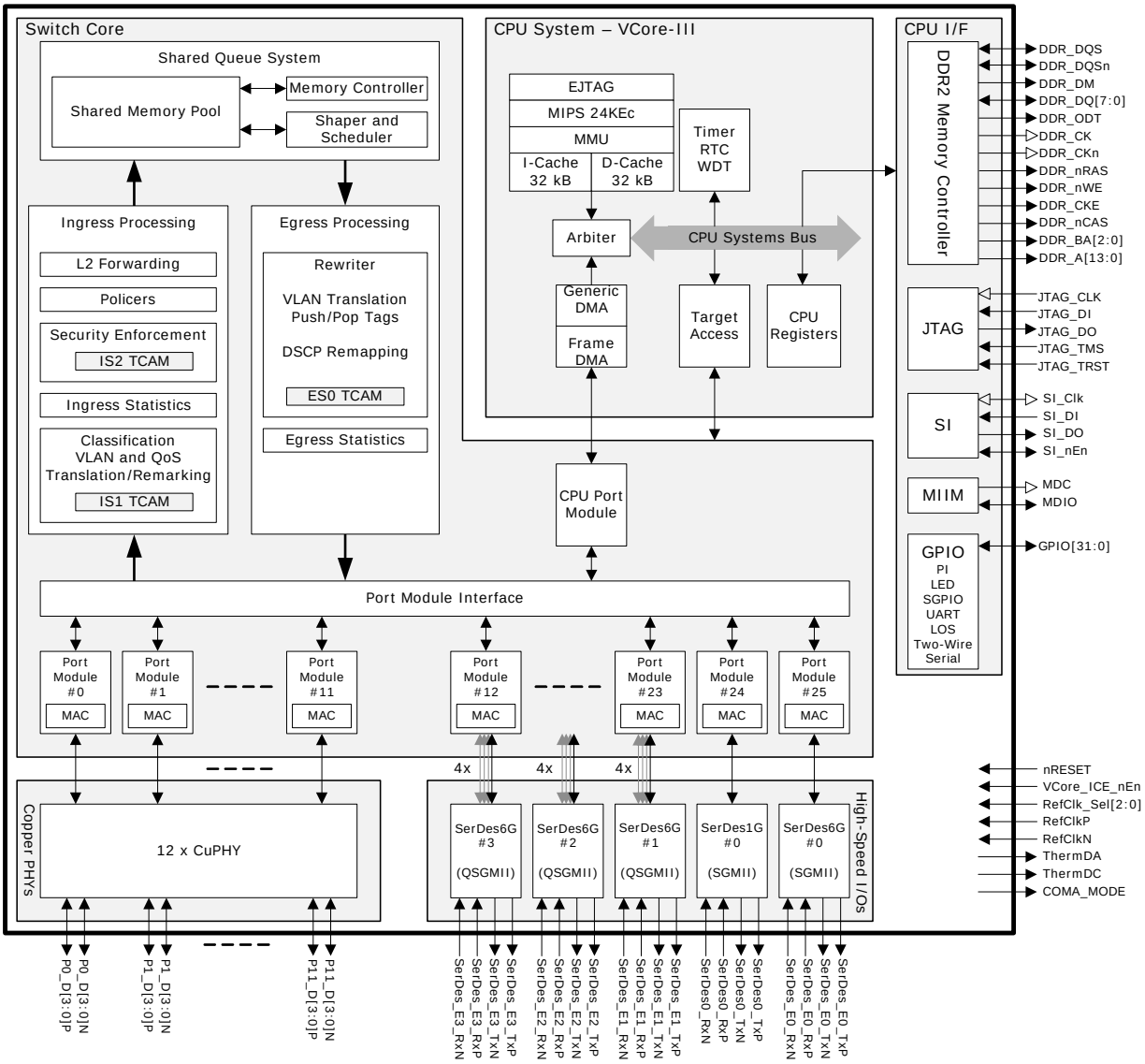
The VSC7420-02, VSC7421-02, and VSC7422-02 family of Layer-2 Ethernet switches provide basic support for QoS and VLAN. They include basic QoS and VLAN classification, as well as a CPU system enabled with a 8051 CPU.

## 3.4 Functional Overview

This section provides an overview all major blocks and functions involved in the bridging operation in the same order as a frame traverses through the devices. It also outlines other major functionality of the device such as the CPU port module, the CPU system, and CPU interfaces.

The following illustration shows the block diagram for the VSC7427-02 device. The other devices in the family have similar block diagrams.

***Figure 1 •*** **VSC7427-02 Block Diagram**




## 3.4.1 Frame Arrival

The Ethernet interfaces receive incoming frames and forward these to the port modules. Supported interfaces include copper transceivers, QSGMII, SGMII, and SerDes.

The integrated low-power copper transceivers support full duplex operation at 10/100/1000 Mbps and half-duplex operation at 10/100 Mbps. The key PHY features are:

- Low power consumption in all modes through ActiPHY™ link down power savings, PerfectReach™ smart cable reach algorithm, and IEEE 802.3az Energy Efficient Ethernet idle power savings.
- VeriPHY® cable diagnostics suite provides extensive network cable operating conditions and status.

There are two programmable direct drive LEDs per port and adjustable brightness levels via register controls with bi-color LED support using both LED pins. The devices also feature a serial LED controller interface for driving LED pins on both internal and external PHYs.

The SGMII ports support both 100BASE-X and 1000BASE-X-SERDES. Each port module contains a Media Access Controller (MAC) that performs a full suite of checks, such as VLAN Tag-aware frame size checking, frame check sequence (FCS) checking, and pause frame identification.

Each port module connecting to a SerDes macro contains a Physical Coding Sublayer (PCS) which perform 8 bits/10 bits encoding, auto-negotiation of link speed and duplex mode, and monitoring of the link status.

Full-duplex is supported for all speeds, and half-duplex is supported for 10 Mbps and 100 Mbps. Symmetric and asymmetric pause flow control are both supported.

All Ethernet ports support Energy Efficient Ethernet (EEE) according to IEEE 802.3az. The shared queue system is capable of controlling the operating states, active or low-power, of the PCS or the internal PHYs. Both the PCS and PHYs understand the line signaling as required for EEE. This includes signaling of active, sleep, quiet, refresh, and wake.

Each QSGMII port can multiplex four port modules onto one I/O interface. Each of the underlying port modules has its own MAC and PCS and can negotiate link speed and duplex mode independently of the other port modules.

## 3.4.2 Basic and Advanced Frame Classification

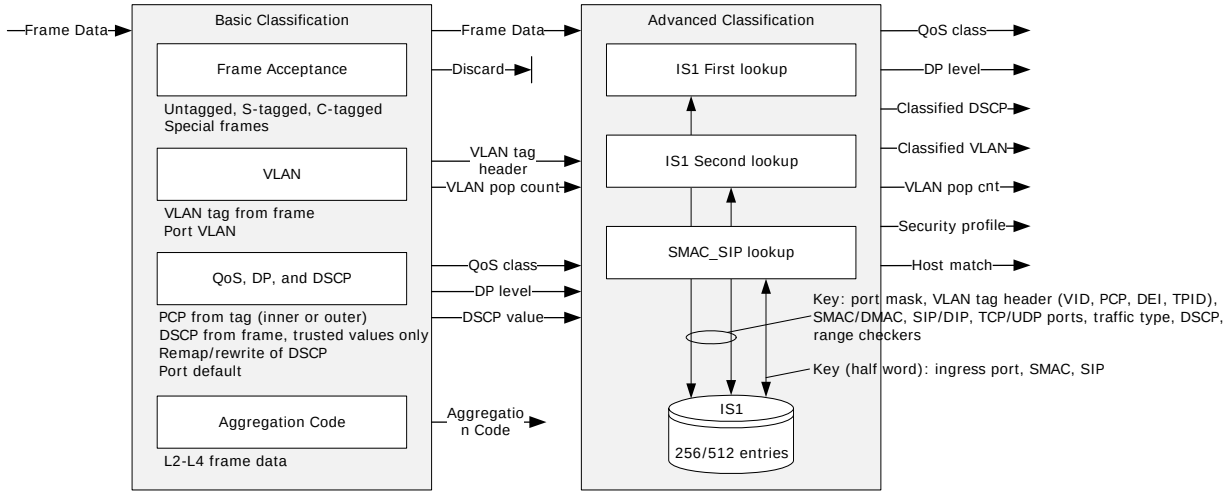
Each frame is sent to the ingress processing module for classification to a VLAN, classification to a Quality of Service (QoS) class, policing, drop precedence marking, collecting statistics, security enforcement, and Layer-2 forwarding.

The classification is a combination of a basic classification using configurable logic and more advanced classification using a TCAM.

The classification engine can understand up to two VLAN tags and can look for Layer-3 and Layer-4 information behind two VLAN tags. If frames are triple tagged, the higher-layer protocol information is not extracted.

The following illustration shows the basic and advanced frame classification.

***Figure 2 •*** **Basic and Advanced Frame Classification**




The basic and advanced classification classifies each frame to a VLAN, a QoS class, DSCP value, and an aggregation code. The basic classification also performs a general frame acceptance check. The output from the basic classification may be overwritten or changed by the more intelligent advanced classification using the IS1 TCAM.

**Frame Acceptance** The frame acceptance filter checks for valid combinations of VLAN tags against the ingress port's VLAN acceptance filter where it is possible to configure rules for accepting untagged, priority-tagged, C-, and S-tagged frames. In addition, the filter also enables discarding of frames with illegal MAC addresses (for instance null MAC address or multicast source MAC address).

**VLAN** Every incoming frame is classified to a VLAN by the basic VLAN classification. This is based on the VLAN in the frame, or if the frame is untagged or the ingress port is VLAN unaware, it is based on the

ingress port's default VLAN. A VLAN classification includes the whole TCI (PCP, DEI, and VID) and also the TPID (C-tag or S-tag).

For double-tagged frames, it is selectable whether the inner or the outer tag is used.

The devices can recognize S-tagged frames with the standard TPID (0x88A8) or S-tagged frames using a custom programmable value. One custom value is supported by the devices.

**QoS and DSCP** Each frame is classified to a Quality of Service (QoS) class. The QoS class is used throughout the devices for providing queuing, scheduling, and congestion control guarantees to the frame according to what is configured for that specific QoS class.

The QoS class is assigned based on the class of service information in the frame's VLAN tags (PCP and DEI) and/or the DSCP values from the IP header. Both IPv4 and IPv6 are supported. If the frame is non-IP or untagged, the port's default QoS class is used.

The DSCP values can be remapped before being used for QoS. This is done using a common table mapping the incoming DSCP to a new value. Remapping is enabled per port. In addition, for each DSCP value, it is possible to specify whether the value is trusted for QoS purposes.

Each IP frame is also classified to an internal DSCP value. By default, this value is taken from the IP header but it may be remapped using the common DSCP mapping table or rewritten based on the assigned QoS class. The classified DSCP value may be written into the frame at egress – this is programmable in the rewriter.

**Aggregation Code** Finally, the basic classification calculates an aggregation code, which is used to select between ports that are member of a link aggregation group. The aggregation code is based on selected Layer-2 through Layer-4 information, such as MAC addresses, IP addresses, IPv6 flow label, and TCP/UDP port numbers. The aggregation code ensures that frames belonging to the same conversation are using the same physical ports in a link aggregation group.

**Advanced Classification**

Following the basic classification, Layer-2 and Layer-4 information is extracted from each frame and matched against a TCAM, IS1, with any mix of up to 256 complex entries (QoS and VLAN) or up to 512 simple entries (host identity check).

The TCAM embeds powerful protocol awareness for well-known protocols such as LLC, SNAP, ARP, IPv4, IPv6, and UDP/TCP. For each frame, three keys are generated and matched against the TCAM. The first two matches are QoS and VLAN relevant, and the last match is a host identity check validating that the frame contains a valid combination of source MAC address and source IP address.

The actions associated with each entry (programmed into the TCAM action RAM) for the first two matches include the ability to overwrite or translate the classified VLAN, overwrite the priority code point (PCP) or the drop eligibility indicator (DEI), overwrite the QoS class and DP level, or overwrite the DSCP value. Each of these actions is enabled individually.

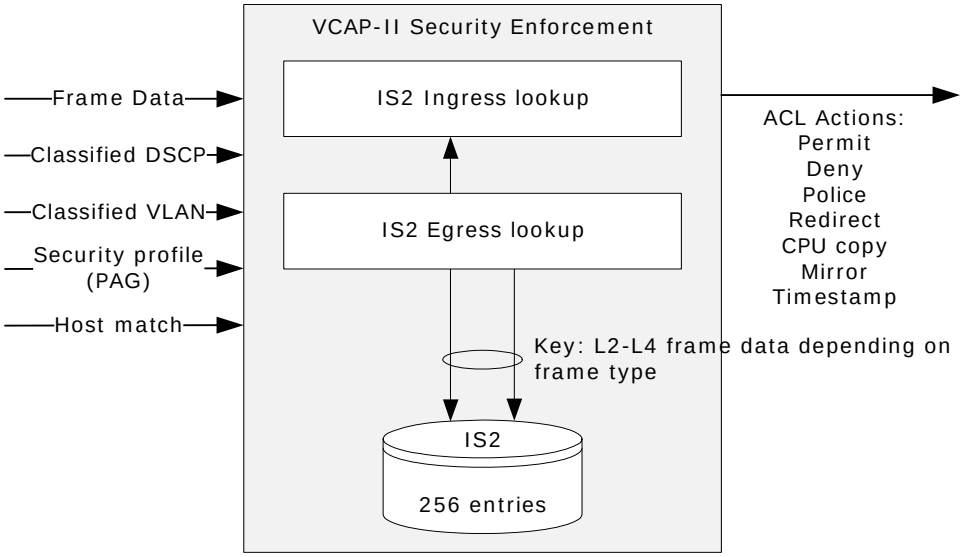
In addition, a policy association group (PAG) is assigned to the frame. The PAG identifies a security profile to which the frame belongs. The PAG is used in the succeeding security frame processor, IS2, to select which access control lists to apply to the frame. The PAG enables creating efficient ACLs that only are applicable to frames with the same PAG.

The host identity validation results in a flag being passed on to the security frame processor IS2 where associated actions such as permit/deny can be programmed.

### 3.4.3 VCAP-II Vitesse Content Aware Processor

All frames are inspected by the VCAP-II IS2 before they are passed on to the Layer-2 forwarding.

***Figure 3 •*** **VCAP-II Security Enforcement**




The VCAP uses a TCAM-based frame processor enabling implementation of a rich set of security features. The flexible VCAP engine supports wire-speed frame inspection based on Layer 2-4 frame information, including the ability to perform longest prefix matching and identifying port ranges. The action associated with each VCAP entry (programmed into the VCAP action RAM) includes the ability to do frame filtering, rate limitation, snooping, redirection, mirroring, and accounting. Even though the VCAP is located in the ingress path of the device, it possesses both ingress and egress capabilities.

The VCAP embeds powerful protocol awareness for well-known protocols such as LLC, SNAP, ARP, IPv4, IPv6, and UDP/TCP.

## 3.4.4 Policing

Each frame is subject to a number of different policing operations. The devices feature a pool of 256 programmable policers. Each frame can trigger three policers from the pool. The pool of policers is split into the followings groups:

- Queue policers: Ingress port number and QoS class determine which policer to use.
- Port policers: Ingress port number determines which policer to use.
- VCAP-II IS2 policers: IS2 action can point to any of the policers in the pool.

It is programmable per port whether to use a port policer or use the queue policers. Policers not used by the port or the queues are available as VCAP-II IS2 policers. It is also programmable whether the policers are working in serial or in parallel.

Each frame is counted in associated statistics reflecting the ingress port and the QoS class. The statistics can count bytes or frames.

Finally, the analyzer contains a group of storm control policers that are capable of policing various kinds of flooding traffic as well as CPU directed learn traffic. These policers are global policers working on all frames received by the switch.

All policers can measure frame rates or bit rates.

## 3.4.5 Layer-2 Forwarding

After the policers, the Layer-2 forwarding block (the analyzer) handles all fundamental bridging operations and maintains the associated MAC table, the VLAN table, and the aggregation table. The devices implement an 8K MAC table and a 4K VLAN table.

The main task of the analyzer is to determine the destination port set of each frame. This forwarding decision is based on various information such as the frame's ingress port, source MAC address, destination MAC address, and the VLAN identifier, as well as the frame's VCAP action, mirroring, and the destination port's link aggregation configuration.

The switch performs Layer-2 forwarding of frames. For unicast and Layer-2 multicast frames, this means forwarding based on the destination MAC address and the VLAN. For IPv4 multicast frames, the switch performs Layer-2 forwarding, but based on Layer-3 information, such as the source IP address. The latter enables source-specific IPv4 multicast forwarding (IGMPv3).

The following describes some of the contributions to the Layer-2 forwarding:

- VLAN classification  VLAN-based forward filtering include source port filtering, destination port filtering, VLAN mirroring, asymmetric VLANs, and so on.
- Security enforcement  The security decision made by the VCAP-II can, for example, redirect the frame to the CPU based on some abnormality detection filters.
- MAC addresses  Destination and source MAC address lookups in the MAC table determine if a frame is a learn frame, a flood frame, a multicast frame, or a unicast frame.
- Learning  By default, the devices perform wire-speed learning on all ports. However, certain ports could be configured with secure learning enabled, where an incoming frame with unknown source MAC address is classified as a “learn frame” and is redirected to the CPU. The CPU performs the learning decision and also decides whether the frame is forwarded.

  Learning can also be disabled. In that case, it does not matter if the source MAC address is in the MAC table.
- Link aggregation  A frame targeted at a link aggregate is further processed to determine which of the link aggregate group ports the frame must be forwarded to.
- Mirroring  Mirror probes may be set up in different places in the forwarding path for monitoring purposes. As part of a mirror a copy of the frame is sent either to the CPU or to another port.

## 3.4.6 Shared Queue System and Egress Scheduler

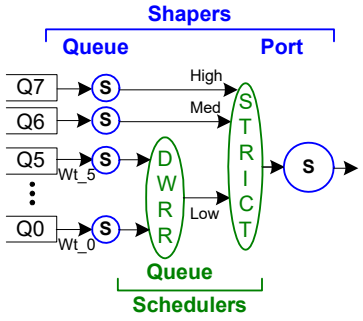
The analyzer provides the destination port set of a frame to the shared queue system. It is the queue system's task to control the frame forwarding to all destination ports.

The shared queue system embeds 4Mbits of memory that can be shared between all queues and ports. The queue system implements egress queues per priority per ingress port. The sharing of resources between queues and ports is controlled by an extensive set of thresholds.

The overall frame latency through the switch is low due to the shared queue system only storing the frame once.

Each egress port implements a scheduler and shapers as shown in the following illustration. Per egress port, the scheduler sees the outcome of aggregating the egress queues (one per ingress port per Qos class) into eight queues, one queue per QoS class. The aggregation is done in a round-robin fashion per QoS class serving all ingress ports equally.

***Figure 4 •*** **Egress Scheduler and Shaper**




When transmitting frames from the shared queue system out on an egress port, frames are scheduled within the port using one of two methods:

- Strict priority – frames with the highest priority are always transmitted before frames with lower priority.
- Deficit Weighted Round Robin (DWRR) – queues 6 and 7 are always strict, and queues 0 through 5 are weighted. Each queue sets a weight ranging from 0 to 31.

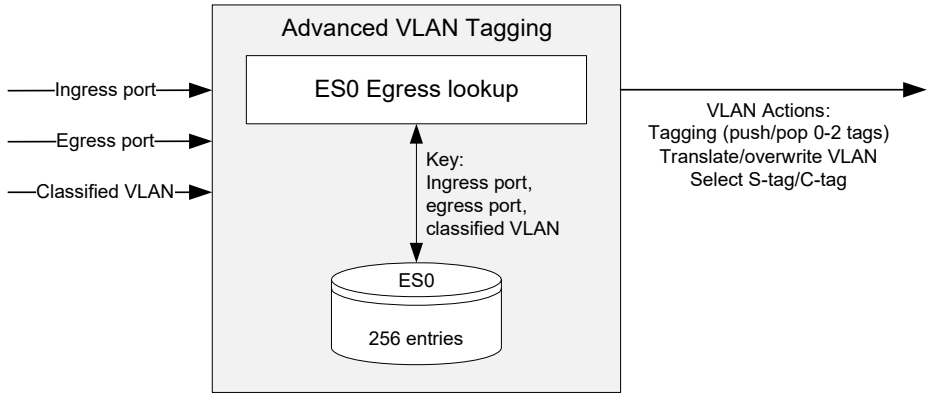
In addition, each egress port implements shapers, one per egress queue and one per port.

## 3.4.7 Rewriter and Frame Departure

Before transmitting the frame on the egress line, the rewriter can modify selected fields in the frame, such as VLAN tags, DSCP value, and FCS.

The rewriter controls the final VLAN tagging of frames based on the classified VLAN, the VLAN pop count, and egress-determined VLAN actions. The egress VLAN actions are by default given by the egress port settings. These include normal VLAN operations such as pushing a VLAN tag, untagging for specific VLANs, and simple translations of DEI and PCP.

***Figure 5 •*** **Advanced VLAN Tagging**




By using the egress TCAM, ES0, much more advanced VLAN tagging operations can be achieved. ES0 enables pushing up to two VLAN tags and allows for a flexible translation of the VLAN tag header. The key into ES0 is the combination of the ingress port, the egress port, and the classified VLAN tag header.

The PCP and DEI bits in the VLAN tag are subject to remarking based on translating the classified tag header or by using the classified QoS value and the frame's drop precedence level from ingress.

In addition, the DSCP value in IP frames can be updated using the classified DSCP value and the frame's drop precedence level from ingress. The DSCP value can be remapped at egress before writing it into the frame.

Finally, the rewriter updates the FCS if the frame was modified before the frame is transmitted.

The egress port module controls the flow control exchange of pause frames with a neighboring device when the interconnection link operates in full-duplex flow control mode. When the connected device triggers flow control through transmission of a pause frame, the MAC stops the egress scheduler's forwarding of frames out of the port. Traffic then builds up in the queue system but sufficient queuing is available to ensure wire speed lossless operation.

In half-duplex operation, the port module's egress path responds to back pressure generation from a connected device by collision detection and frame retransmission.

## 3.4.8 CPU Port Module

The CPU port module contains eight CPU extraction queues and two CPU injection queues. These queues provide an interface for exchanging frames between the internal CPU system and the switch core. An external CPU using the serial interface can also inject and extract frames to and from the switch core by using the CPU port module. Additionally, any Ethernet interface on the devices can be used for extracting and injecting frames.

The switch core can intercept a variety of different frame types and copy or redirect these to the CPU extraction queues. The classifier can identify a set of well-known frames such as IEEE reserved destination MAC addresses (BPDUs, GARPs), as well as IP-specific frames (IGMP, MLD). The security TCAM, IS2, provides another very flexible way of intercepting all kinds of frames, for instance specific OAM frames, ARP frames or explicit applications based on TCP/UDP port numbers. In addition, frames can be intercepted based on the MAC table, the VLAN table, or the learning process.

Whenever a frame is copied or redirected to the CPU, a CPU extraction queue number is associated with the frame and used by the CPU port module when enqueuing the frame into the 8 CPU extraction queues. The CPU extraction queue number is programmable for every interception option in the switch core.

### 3.4.9 CPU System and Interfaces

The devices feature a VCore-III CPU system containing a powerful 416 MHz MIPS 24KEc™ CPU. It is suitable for lightly managed and fully managed applications.

VCore-III includes a general-purpose direct memory access engine (GPDMA) that also supports frame-based direct memory access (FMDA) operations. The FMDA offloads the CPU when injecting and extracting frames to and from the switch core. VCore-III boots up from a serial flash and uses DDR2 SDRAM for memory, in addition to its built-in 32 kilobytes of instruction cache and 32 kilobytes of data cache. An external debugger can be attached to the EJTAG interface.

In addition to the integrated processor, the CPU system permits the attachment of an external CPU. For configuration of switch register, an external CPU can use either a serial interface or an MII Management interface. For frame transfers, the external CPU has the option of using the serial interface, an MII Management interface, or an SGMII port.

The devices include a GPIO interface with 32 individually configurable pins. Through the GPIOs, various interfaces are supported by the devices:

- Two-wire serial interface (two GPIO pins)
- Eight-bit parallel interface (sixteen GPIO pins)
- UART (two GPIO pins)
- External interrupts (two interrupt pins)
- Serial GPIO (SGPIO) and LED interface (four GPIO pins)
- Fan controller with speed input and pulse-width-modulated output (two GPIO pins)
- MII Management slave interface for accessing switch registers from an external CPU (two GPIO pins)
- Direct drive LEDs (two pins per internal PHY)

The Serial GPIO and LED interface can specifically be used for driving external LEDs for the internal and external copper PHYs or for serializing external interrupts, for instance link down events from external PHYs, before being input to the devices.

Finally, each of the devices has two MII management controllers; one for the internal PHYs and one connected to the MIIM interface for controlling external PHYs.

# 4 Functional Descriptions

This section provides detailed information about the functional aspects of the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Managed Gigabit Ethernet switch devices, available configurations, operational features, and testing functionality.

## 4.1 Port Modules

The port modules contain the following functional blocks:

- MAC
- PCS (ports connecting to a high-speed I/O SerDes macro)

Ports connecting to one of the integrated copper transceivers do not have a PCS.

### 4.1.1 Port Module Numbering and Macro Connections

The port modules connect to the interface macros. The interface macros can be of three types:

- Internal copper PHY
- SERDES6G macro
- SERDES1G macro

The interface macros connect to the external interface pins. For more information about the SerDes macros and integrated copper transceivers, see SERDES1G, page 22, SERDES6G, page 26, and Copper Transceivers, page 32. Which switch core port modules are connected to which interface macros depends on part number and for some parts on internal configuration.

VSC7425-02 can be used in two different port configurations: switch mode 0 or switch mode 2. The VSC7424-02, VSC7426-02, and VSC7427-02 devices run in switch mode 0. The switch mode is controlled through DEVCPU_GCB::MISC_CFG.SW_MODE.

The following table lists the mapping from the switch core port modules to the interface macros. Empty cells in the table imply that the port module number is not in use for the specific part number.

When programming registers depending on port numbers, the switch core port module number must always be used. Examples of this are when accessing port module registers (PORT::), using port masks in system or analyzer registers (SYS::, ANA::), or programming VCAP entries with port number information or port masks.

The number next to the interface macro type (for example, “3” in cell SERDES6G, 3) indicates either the macro number or the internal PHY number that must be used when addressing the macros and PHYs for programming.

***Table 3 •*** **Port Mapping from Switch Core Port Module to Interface Macros**

| Switch Core Port Module | VSC7424-02 | VSC7425-02 Switch Mode 0 | VSC7425-02 Switch Mode 2 | VSC7426-02 | VSC7427-02 |
|---|---|---|---|---|---|
| 0-7 | CuPHY, 0-7 | CuPHY, 0-7 | CuPHY, 0-7 | CuPHY, 0-7 | CuPHY, 0-7 |
| 8-11 | | CuPHY, 8-11 | CuPHY, 8-11 | CuPHY, 8-11 | CuPHY, 8-11 |
| 12-15 | | | | SERDES6G, 3 | SERDES6G, 3 |
| 16-19 | | | | SERDES6G, 2 | SERDES6G, 2 |
| 20 | | SERDES6G, 1 | SERDES1G, 5 | SERDES6G, 1 | SERDES6G, 1 |
| 21 | | SERDES6G, 1 | SERDES1G, 4 | SERDES6G, 1 | SERDES6G, 1 |
| 22 | | SERDES6G, 1 | SERDES1G, 3 | SERDES6G, 1 | SERDES6G, 1 |
| 23 | | SERDES6G, 1 | SERDES1G, 2 | SERDES6G, 1 | SERDES6G, 1 |

*Table 3 •* **Port Mapping from Switch Core Port Module to Interface Macros**

| Switch Core Port Module | VSC7424-02 | VSC7425-02 Switch Mode 0 | VSC7425-02 Switch Mode 2 | VSC7426-02 | VSC7427-02 |
|---|---|---|---|---|---|
| 24 | SERDES1G, 0 | SERDES1G, 0 | SERDES1G, 0 | | SERDES1G, 0 |
| 25 | SERDES1G, 1 | SERDES6G, 0 | SERDES1G, 1 | | SERDES6G, 0 |
| 26 | CPU port | CPU port | CPU port | CPU port | CPU port |

## 4.1.2 MAC

This section provides information about the high-level functionality and the configuration options of the Media Access Controller (MAC) that is used in each of the port modules.

The MAC supports the following speeds and duplex modes:

- PHY ports support 10/100/1000 Mbps in full-duplex mode and 10/100 Mbps in half-duplex mode.
- SERDES1G port support 10/100/1000 Mbps in full-duplex mode and 10/100 Mbps in half-duplex mode.

The following table lists the registers associated with configuring the MAC.

*Table 4 •* **MAC Configuration Registers**

| Register | Description | Replication |
|---|---|---|
| CLOCK_CFG | Reset and speed configuration | Per port |
| MAC_ENA_CFG | Enabling of Rx and Tx data paths | Per port |
| MAC_MODE_CFG | Port mode configuration | Per port |
| MAC_MAXLEN_CFG | Maximum length configuration | Per port |
| MAC_TAGS_CFG | VLAN tag length configuration | Per port |
| MAC_ADV_CHK_CFG | Type length configuration | Per port |
| MAC_IFG_CFG | Interframe gap configuration | Per port |
| MAC_HDX_CFG | Half-duplex configuration | Per port |
| MAC_FC_CFG | Flow control configuration | Per port |
| MAC_FC_MAC_LOW_CFG | LSB of SMAC used in pause frames | Per port |
| MAC_FC_MAC_HIGH_CFG | MSB of SMAC used in pause frames | Per port |
| MAC_STICKY | Sticky bit recordings | Per port |

### 4.1.2.1 Resets

There are a number of resets in the port module. All of the resets can be set and cleared simultaneously. By default, all blocks are in the reset state. With reference to register CLOCK_CFG, the resets are:

- MAC_RX_RST — Reset of the MAC receiver
- MAC_TX_RST — Reset of the MAC transmitter
- PORT_RST — Reset of the ingress and egress queues
- PHY_RST — Reset of the integrated PHY (only present for port modules connecting to a PHY)
- PCS_RX_RST — Reset of the PCS decoder (only present for port modules connecting to a SerDes macro)
- PCS_TX_RST — Reset of the PCS encoder (only present for port modules connecting to a SerDes macro)

When changing the MAC configuration, the port must go through a reset cycle. This is done by writing register CLOCK_CFG twice. On the first write, the reset bits are set. On the second write, the reset bits are cleared. Bits that are not reset bits in CLOCK_CFG must keep their new value for both writes.

For more information about resetting a port, see Port Reset Procedure, page 191.

### 4.1.2.2 Port Mode Configuration

The MAC provides a number of handles for configuring the port mode. With reference to the MAC_MODE_CFG, MAC_IFG_CFG, and MAC_ENA_CFG registers, the handles are:

- Duplex mode (FDX_ENA). Half or full duplex.
- Data sampling (GIGA_MODE_ENA). Must be 1 in 1 Gbps and 0 in 10 Mbps and 100 Mbps.
- Enabling transmission and reception of frames (TX_ENA/RX_ENA). Clearing RX_ENA stops the reception of frames and further frames are discarded. An ongoing frame reception is interrupted. Clearing TX_ENA stops the dequeuing of frames from the egress queues, which means that frames are held back in the egress queues. An ongoing frame transmission is completed.
- Tx to Tx inter-frame gap (TX_IFG).

For ports connecting to an internal PHY, the link speed is determined by the PHY. For other ports, the link speed is configured using CLOCK_CFG.LINK_SPEED with the following options:

- Link speed (CLOCK_CFG.LINK_SPEED)
  1 Gbps (125 MHz clock)

  100 Mbps (25 MHz clock)

  10 Mbps (2.5 MHz clock)

### 4.1.2.3 Half-Duplex Mode

A number of special configuration options are available for half-duplex (HDX) mode:

- **Seed for back-off randomizer** Field MAC_HDX_CFG.SEED seeds the randomizer used by the backoff algorithm. Use MAC_HDX_CFG.SEED_LOAD to load a new seed value.
- **Backoff after excessive collision** Field MAC_HDX_CFG.WEXC_DIS determines whether the MAC backs off after an excessive collision has occurred. If set, backoff is disabled after excessive collisions.
- **Retransmission of frame after excessive collision** Field MAC_HDX_CFG.RETRY_AFTER_EXC_COL_ENA determines if the MAC retransmits frames after an excessive collision has occurred. If set, a frame is not dropped after excessive collisions, but the backoff sequence is restarted. Although this is a violation of IEEE 802.3, it is useful in non-dropping half-duplex flow control operation.
- **Late collision timing** Field MAC_HDX_CFG.LATE_COL_POS adjusts the border between a collision and a late collision in steps of 1 byte. According to IEEE 802.3, section 21.3, this border is permitted to be on data byte 56 (counting frame data from 1); that is, a frame experiencing a collision on data byte 55 is always retransmitted, but it is never retransmitted when the collision is on byte 57. For each higher LATE_COL_POS value, the border is moved 1 byte higher.
- **Rx-to-Tx inter-frame gap** The sum of MAC_IFG_CFG.RX_IFG1 and MAC_IFG_CFG.RX_IFG2 establishes the time for the Rx-to-Tx inter-frame gap. RX_IFG1 is the first part of half-duplex Rx-to-Tx inter-frame gap. Within RX_IFG1, this timing is restarted if carrier sense (CRS) has multiple high-low transitions (due to noise). RX_IFG2 is the second part of half-duplex Rx-to-Tx inter-frame gap. Within RX_IFG2, transitions on CRS are ignored.

When enabling a port for half-duplex mode, the switch core must also be enabled (SYS::FRONT_PORT_MODE.HDX_MODE).

### 4.1.2.4 Frame and Type/Length Check

The MAC supports frame lengths of up to 16 kilobytes. The maximum length accepted by the MAC is configurable in MAC_MACLEN_CFG.MAX_LEN.

The MAC allows tagged frames to be 4 bytes longer and double-tagged frames to be 8 bytes longer than the specified maximum length (MAC_TAGS_CFG.VLAN_LEN_AWR_ENA). The MAC must be configured to look for VLAN tags. By default, EtherType 0x8100 identifies a VLAN tag. In addition, a custom EtherType can be configured in MAC_TAGS_CFG.TAG_ID. The MAC can be configured to look for none, one, or two tags (MAC_TAG_CFG.VLAN_AWR_ENA, MAC_TAG_CFG.VLAN_DBL_AWR_ENA).

The type/length check (MAC_ADV_CHK_CFG.LEN_DROP_ENA) causes the MAC to discard frames with type/length errors (in-range and out-of-range errors).

#### 4.1.2.5 Flow Control

In full-duplex mode, the MAC provides independent support for transmission of pause frames and reaction to incoming pause frames. This allows for asymmetric flow control configurations.

The MAC obeys received pause frames (MAC_FC_CFG.RX_FC_ENA) by pausing the egress traffic according to the timer values specified in the pause frames.

The transmission of pause frames is triggered by assertion of a flow control condition in the ingress queues caused by a queue filling exceeding a watermark. For more information, see Shared Queue System, page 104. The MAC handles the formatting and transmission of the pause frame. The following configuration options are available:

- Transmission of pause frames (MAC_CFG_CFG.TX_FC_ENA).
- Pause timer value used in transmitted pause frames (MAC_FC_CFG.PAUSE_VAL_CFG).
- Flow control cancellation when the ingress queues de-assert the flow control condition by transmission of a pause frame with timer value 0 (MAC_FC_CFG.ZERO_PAUSE_ENA).
- Source MAC address used in transmitted pause frames (MAC_FC_MAC_HIGH_CFG, MAC_FC_MAC_LOW_CFG).

The MAC has the option to discard incoming frames when the remote link partner is not obeying the pause frames transmitted by the MAC. The MAC discards an incoming frame if a Start-of-Frame is seen after the pause frame was transmitted. It is configurable how long reaction time is given to the link partner (MAC_FC_CFG.FC_LATENCY_CFG). The benefit of this approach is that the queue system is not risking being overloaded with frames due to a non-complying link partner.

In half-duplex mode, the MAC does not react to received pause frames. If the flow control condition is asserted by the ingress queues, the industry-standard backpressure mechanism is used. Together with the ability to retransmit frames after excessive collisions (MAC_HDX_CFG.RETRY_AFTER_EXC_COL_ENA), this enables non-dropping half-duplex flow control.

#### 4.1.2.6 Frame Aging

The following table lists the registers associated with frame aging.

***Table 5 •*** **Frame Aging Configuration Registers**

| Register | Description | Replication |
|---|---|---|
| SYS::FRM_AGING | Frame aging time | None |
| REW::PORT_CFG.AGE_DIS | Disable frame aging | Per port |

The MAC supports frame aging where frames are discarded if a maximum transit delay through the switch is exceeded. All frames, including CPU-injected frames, are subject to aging. The transit delay is time from when a frame is fully received until that frame is scheduled for transmission through the egress MAC. The maximum allowed transit delay is configured in SYS::FRM_AGING.

Frame aging can be disabled per port (REW::PORT_CFG.AGE_DIS).

Discarded frames due to frame aging are counted in the c_tx_aged counter.

### 4.1.3 PCS

This section provides information about the Physical Coding Sublayer (PCS) block, where the auto-negotiation process establishes mode of operation for a link. The PCS supports both SGMII mode and two SerDes modes, 1000BASE-X and 100BASE-FX.

The PCS block is only available in port modules 12 through 25.

The following table lists the registers associated with PCS.

***Table 6 •*** **PCS Configuration Registers**

| Registers | Description | Replication |
|---|---|---|
| PCS1G_CFG | PCS configuration | Per PCS |
| PCS1G_MODE_CFG | PCS mode configuration | Per PCS |
| PCS1G_SD_CFG | Signal detect configuration | Per PCS |
| PCS1G_ANEG_CFG | Configuration of the PCS auto-negotiation process | Per PCS |
| PCS1G_ANEG_NP_CFG | Auto-negotiation next page configuration | Per PCS |
| PCS1G_LB_CFG | Loop-back configuration | Per PCS |
| PCS1G_ANEG_STATUS | Status signaling of the PCS auto-negotiation process | Per PCS |
| PCS1G_ANEG_NP_STATUS | Status signaling of the PCS auto-negotiation next page process | Per PCS |
| PCS1G_LINK_STATUS | Link status | Per PCS |
| PCS1G_LINK_DOWN_CNT | Link down counter | Per PCS |
| PCS1G_STICKY | Sticky bit register | Per PCS |

The PCS is enabled in PCS1G_CFG.PCS_ENA and supports both SGMII and 1000BASE-X SERDES mode (PCS_MODE_CFG.SGMII_MODE_ENA), as well as 100-BASE-FX. For information about enabling 100BASE-FX, see 100BASE-FX, page 21.

The PCS also supports the IEEE 802.3, Clause 66 unidirectional mode, where the transmission of data is independent of the state of the receive link (PCS_MODE_CFG.UNIDIR_MODE_ENA).

#### 4.1.3.1 Auto-Negotiation

Auto-negotiation is enabled in PCS1G_ANEG_CFG.ANEG_ENA. To restart the auto-negotiation process, PCS1G_ANEG_CFG.ANEG_RESTART_ONE_SHOT must be set.

In SGMII mode (PCS_MODE_CFG.SGMII_MODE_ENA=1), matching the duplex mode with the link partner must be ignored (PCS1G_ANEG_CFG.SW_RESOLVE_ENA). Otherwise, the link is kept down when the auto-negotiation process fails.

The advertised word for the auto-negotiation process (base page) is configured in PCS1G_ANEG_CFG.ADV_ABILITY. The next page information is configured in PCS1G_ANEG_NP_CFG.NP_TX.

When the auto-negotiation state machine has exchanged base page abilities, the PCS1G_ANEG_STATUS.PAGE_RX_STICKY is asserted indicating that the link partner's abilities were received (PCS1G_ANEG_STATUS.LP_ADV_ABILITY).

If next page information is exchanged, PAGE_RX_STICKY must be cleared, next page abilities must be written to PCS1G_ANEG_NP_CFG.NP_TX, and PCS1G_ANEG_NP_CFG.NP_LOADED_ONE_SHOT must be set. When the auto-negotiation state machine has exchanged the next page abilities, the PCS1G_ANEG_STATUS.PAGE_RX_STICKY is asserted again, indicating that the link partner's next page abilities were received (PCS1G_ANEG_STATUS.LP_NP_RX). Additional exchanges of next page information are possible using the same procedure.

After the last next page is received, the auto-negotiation state machine enters the IDLE_DETECT state and the PCS1G_ANEG_STATUS.PR bit is set indicating that ability information exchange (base page and possible next pages) is finished and software can now resolve priority. Appropriate actions, such as

Rx or Tx reset, or auto-negotiation restart, can then be taken, based on the negotiated abilities. The LINK_OK state is reached one link timer period later.

When the auto-negotiation process reaches the LINK_OK state, PCS1G_ANEG_STATUS.ANEG_COMPLETE is asserted.

### 4.1.3.2 Link Surveillance

The current link status can be observed through PCS1G_LINK_STATUS.LINK_STATUS. The LINK_STATUS is defined as either the PCS synchronization state or as bit 15 of PCS1G_ANEG_STATUS.LP_ADV_ABILTY, which carries information about the link status of the attached PHY in SGMII mode.

Link down is defined as the auto-negotiation state machine being in neither the AN_DISABLE_LINK_OK state nor the LINK_OK state for one link timer period. If a link down event occurs, PCS1G_STICKY.LINK_DOWN_STICKY is set, and PCS1G_LINK_DOWN_CNT is incremented. In SGMII mode, the link timer period is 1.6 ms; in SerDes mode, the link timer period is 10 ms.

The PCS synchronization state can be observed through PCS1G_LINK_STATUS.SYNC_STATUS. Synchronization is lost when the PCS is not able to recover and decode data received from the attached serial link.

### 4.1.3.3 Signal Detect

The PCS can be enabled to react to loss of signal through signal detect (PCS1G_SD_CFG.SD_ENA). At loss of signal, the PCS Rx state machine is restarted, and frame reception stops. If signal detect is disabled, no action is taken upon loss of signal. The polarity of signal detect is configurable in PCS1G_SD_CFG.SD_POL.

The source of signal detect is selected in PCS1G_SD_CFG.SD_SEL to either the SerDes PMA or the PMD receiver. If the SerDes PMA is used as source, the SerDes macro provides the signal detect. If the PMD receiver is used as source, signal detect is sampled externally through one of the GPIO pins on the devices. For more information about the configuration of the GPIOs and signal detect, see GPIO Controller, page 176.

PCS1G_LINK_STATUS.SIGNAL_DETECT contains the current value of the signal detect input.

### 4.1.3.4 Tx Loopback

For debug purposes, the Tx data path in the PCS can be looped back into the Rx data path. This feature is enabled through PCS1G_LB_CFG.TBI_HOST_LB_ENA.

### 4.1.3.5 Test Patterns

The following table lists the registers associated with configuring test patterns.

***Table 7 •*** **Test Pattern Registers**

| Registers | Description | Replication |
|---|---|---|
| PCS1G_TSTPAT_MODE_CFG | Test pattern configuration | Per PSC |
| PCS1G_TSTPAT_MODE_STATUS | Test pattern status | Per PCS |

PCS1G_TSTPAT_MODE_CFG.JTP_SEL overwrites normal operation of the PCS and enables generation of jitter test patterns for debugging. The jitter test patterns are defined in IEEE 802.3, Annex 36A, and the following patterns are supported:

- High frequency test pattern
- Low frequency test pattern
- Mixed frequency test pattern
- Continuous random test pattern with long frames
- Continuous random test pattern with short frames

PCS1G_TSTPAT_MODE_STATUS register holds information about error and lock conditions while running the jitter test patterns.

### 4.1.3.6 Low Power Idle

The following table lists the registers associated with low power idle (LPI).

***Table 8 •*** **Low Power Idle Registers**

| Registers | Description | Replication |
|---|---|---|
| PCS1G_LPI_CFG | Configuration of the PCS Low Power Idle process | Per PSC |
| PCS1G_LPI_WAKE_ERROR_CNT | Error counter | Per PCS |
| PCS1G_LPI_STATUS | Low Power Idle status | Per PCS |

The PCS supports Energy Efficient Ethernet (EEE) as defined by IEEE 802.3az. The PCS converts Low Power Idle (LPI) encoding between the MAC and the serial interface transparently. In addition, the PCS provides control signals allowing to stop data transmission in the SerDes macro. During low power idles the serial transmitter in the SerDes macro can be powered down, only interrupted periodically while transmitting refresh information, which allows the receiver to notice that the link is still up but in power down mode.

When a SERDES6G macro operating in QSGMII mode is enabled for powering down of the serial transmitter during low power idles, one of the four PCSs connected to the macro must be selected master (PCS1G_LPI_CFG.QSGMII_MS_SEL). The master PCS sends refresh information to the attached receivers periodically. Note that the serial transmitter can only power down when all four attached ports are in low power idle.

For more information about powering down the serial transmitter in the SerDes macros, see SERDES1G, page 22 or SERDES6G, page 26.

It is not necessary to enable the PCS for EEE, because it is controlled indirectly by the shared queue system. It is possible, however, to manually force the PCS into the low power idle mode through PCS1G_LPI_CFG.TX_ASSERT_LPIDLE. During LPI mode, the PCS constantly encodes low power idle with periodical refreshes. For more information about EEE, see Energy Efficient Ethernet, page 111.

The current low power idle state can be observed through PCS1G_LPI_STATUS for both receiver and transmitter:

- RX_LPI_MODE: Set if the receiver is in low power idle mode.
- RX_QUIET: Set if the receiver is in the Quiet state of the low power idle mode. If cleared while RX_LPI_MODE is set, the receiver is in the refresh state of the low power idle mode.

The same is observable for the transmitter through TX_LPI_MODE and TX_QUIET.

If an LPI symbol is received, the RX_LPI_EVENT_STICKY bit is set, and if an LPI symbol is transmitted, the TX_LPI_EVENT_STICKY bit is set. These events are sticky.

The PCS1G_LPI_WAKE_ERROR_CNT wake-up error counter increments when the receiver detects a signal and the PCS is not synchronized. This can happen when the transmitter fails to observe the wake-up time or if the receiver is not able to synchronize in time.

### 4.1.3.7 100BASE-FX

The following table lists the registers associated with 100BASE-FX configuration.

***Table 9 •*** **100BASE-FX Registers**

| Registers | Description | Replication |
|---|---|---|
| PCS_FX100_CFG | Configuration of the PCS 100BASE-FX mode | Per PSC |
| PCS_FX100_STATUS | Status of the PCS 100BASE-FX mode | Per PCS |

The PCS supports a 100BASE-FX mode in addition to the SGMII and 1000BASE-X SerDes modes. The 100BASE-FX mode uses 4-bit/5-bit coding as specified in IEEE 802.3 Clause 24 for fiber connections. The 100BASE-FX mode is enabled through PCS_FX100_CFG.PCS_ENA, which masks out all PCS1G related registers.

The following options are available:

**Far-End Fault facility** In 100BASE-FX, the PCS supports the optional Far-End Fault facility. Both Far-End Fault generation (PCS_FX100_CFG.FEF_GEN_ENA) and Far-End Fault Detection (PCS_FX100_CFG.FEF_CHK_ENA) are supported. An Far-End Fault incident is recorded in PCS_FX100_STATUS.FEF_FOUND.

**Signal Detect** 100BASE-FX has a similar signal detect scheme to the SGMII and SerDes modes. For 100BASE-FX, PCS_FX100_CFG.SD_ENA enables signal detect, PCS_FX100_CFG.SD_POL controls the polarity, and PCS_FX100_CFG.SD_SEL selects the input source. The current status of the signal detect input can be observed through PCS_FX100_STATUS.SIGNAL_DETECT. For more information about signal detect, see Signal Detect, page 20.

**Link Surveillance** The PCS synchronization status can be observed through PCS_FX100_STATUS.SYNC_STATUS. When synchronization is lost, the link breaks and PCS_FX100_STATUS.SYNC_LOST_STICKY is set. The PCS continuously tries to recover the link.

**Unidirectional mode** 100BASE-FX has a similar unidirectional mode as SGMII and SerDes modes. PCS_FX100_CFG.UNIDIR_MODE_ENA enables unidirectional mode.

## 4.2 SERDES1G

SERDES1G is a high-speed SerDes interface that operates at 1 Gbps (SGMII/SerDes) and 100 Mbps (100BASE-FX). The 100BASE-FX mode is supported by oversampling.

The following table lists the registers associated with SERDES1G.

***Table 10 •*** **SERDES1G Registers**

| Registers | Description | Replication |
|---|---|---|
| SERDES1G_COMMON_CFG | Common configuration | Per SerDes |
| SERDES1G_DES_CFG | Deserializer configuration | Per SerDes |
| SERDES1G_IB_CFG | Input buffer configuration | Per SerDes |
| SERDES1G_SER_CFG | Serializer configuration | Per SerDes |
| SERDES1G_OB_CFG | Output buffer configuration | Per SerDes |
| SERDES1G_PLL_CFG | PLL configuration | Per SerDes |
| SERDES1G_MISC_CFG | Miscellaneous configuration | Per SerDes |

For increased performance in specific application environments, SERDES1G supports the following:

- Programmable loop-bandwidth and phase regulation of deserializer
- Input buffer signal detect/loss of signal (LOS) options
- Input buffer with equalization
- Programmable output buffer features, including:

  De-emphasis

  Amplitude drive levels

  Slew rate control

  Idle mode
- Synchronous Ethernet support
- Loopbacks for system test

## 4.2.1 SERDES1G Basic Configuration

The SERDES1G is enabled in SERDES1G_COMMON_CFG.ENA_LANE. By default, the SERDES1G is held in reset and must be released before the interface is active. This is done through SERDES1G_COMMON_CFG.SYS_RST and SERDES1G_MISC_CFG.LANE_RST.

### 4.2.1.1 SERDES1G PLL Frequency Configuration

To operate the SERDES1G block at 1.25 GHz (corresponding to 1 Gbps data rate), configure the internal macro PLL as follows:

1. Configure SERDES1G_PLL_CFG.PLL_FSM_CTRL_DATA to 200.
2. Set SYS_RST = 0 (active) and PLL_FSM_ENA = 0 (inactive).
3. Set SYS_RST = 1 (deactive) and PLL_FSM_ENA = 1 (active).

## 4.2.2 SERDES1G Loopback Modes

The SERDES1G interface supports two different loopback modes for testing and debugging data paths: equipment loopback and facility loopback.

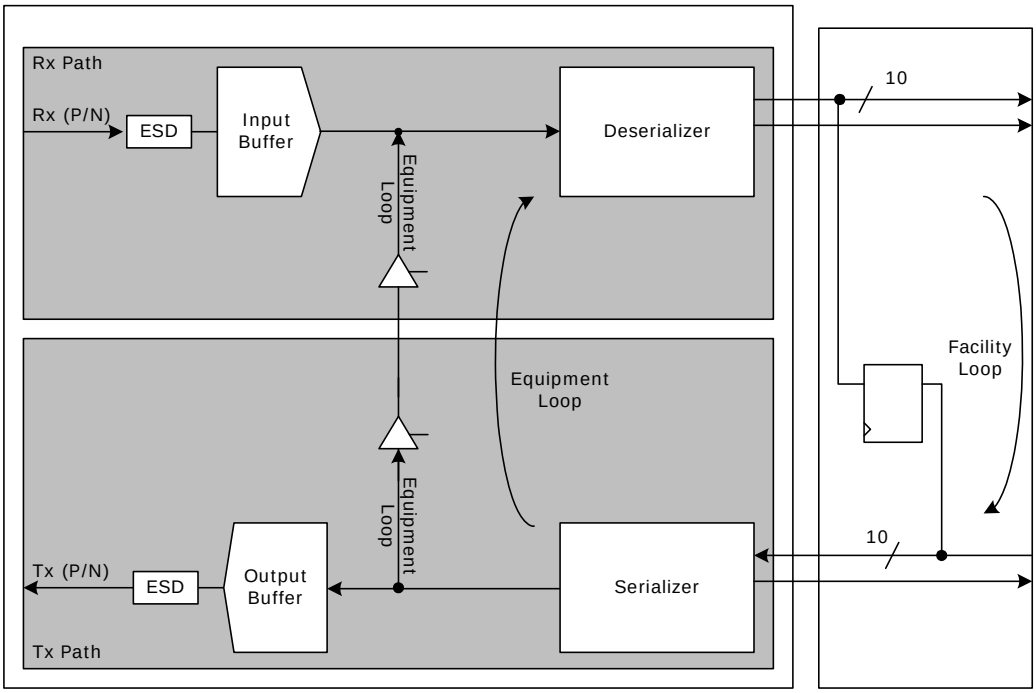
**Equipment loopback (SERDES1G_COMMON_CFG.ENA_ELOOP)** Data is looped back from serializer output to deserializer input, and the receive clock is recovered. The equipment loopback includes all transmit and receive functions, except for the input and output buffers. The Tx data can still be observed on the output.

**Facility loopback (SERDES1G_COMMON_CFG.ENA_FLOOP)** The clock and parallel data output from deserializer are looped back to the serializer interface. Incoming serial data passes through the input buffer, the CDR, the deserializer, back to the serializer, and finally out through the output buffer.

Only one of the loopbacks can be enabled at the same time.

The following illustration shows the loopback paths.

***Figure 6 •*** **SERDES1G Loopback Modes**




## 4.2.3 SERDES1G Deserializer Configuration

The SERDES1G block includes digital control logic that interacts with the analog modules within the block and compensates for the frequency offset between the received data and the internal high-speed

reference clock. To gain high jitter performance, the phase regulation is a PI-type regulator, whose proportional (P) and integrative (I) characteristics can be independently configured.

The integrative part of the phase regulation loop is configured in SERDES1G_DES_CFG.DES_PHS_CTRL. The limits of the integrator are programmable, allowing different settings for the integrative regulation while guaranteeing that the proportional part still is stronger than the integrative part. Integrative regulation compensates frequency modulation from DC up to cut-off frequency. Frequencies above the cut-off frequency are compensated by the proportional part.

The time constant of the integrator is controlled independently of the proportional regulation by SERDES1G_DES_CFG.DES_BW_HYST. The DES_BW_HYST register field is programmable in a range from 3 to 7. The lower the configuration setting, the smaller the time-constant of the integrative regulation. For normal operation, configure DES_BW_HYST to 5.

The cut-off-frequency is calculated to:

fco = 1/(2 × PI × 128 × PLL period × 32 × 2^(DES_BW_HYST + 1 – DES_BW_ANA))

PLL period = 1/(data rate)

The integrative regulator can compensate a static frequency offset within the programmed limits down to a remaining frequency error of below 4 ppm. In steady state, the integrator toggles between two values around the exact value, and the proportional part of the phase regulation takes care of the remaining phase error.

After a device reset, the phase regulation may be 180° out of phase compared to the incoming data, resulting in a deadlock condition at the sampling stage of the deserializer. To prevent this situation, the SERDES1G provides a 180° deadlock protection mechanism (SERDES1G_DES_CFG.DES_MBTR_CTRL). If the deadlock protection mechanism is enabled, a small frequency offset is applied to the phase regulation loop. The offset is sufficient to move the sampling point out of the 180° deadlock region, while at the same time, small enough to allow the regulation loop to compensate when the sample point is within the data eye.

The loop bandwidth for the proportional part of the phase regulation loop is controlled by configuring SERDES1G_DES_CFG.DES_BW_ANA.

The fastest loop bandwidth setting (lowest configuration value) results in a loop bandwidth that is equal to the maximum frequency offset compensation capability. For improved jitter performance, use a setting with sufficient margin to track the expected frequency offset rather than using the maximum frequency offset. For example, if a 100 ppm offset is expected, use a setting that is four times higher than the offset. For more information about possible bandwidth selections, see Table 503, page 396 and Table 516, page 404.

The following table provides the limits for the frequency offset compensation. The values are theoretical limits for input signals without jitter, because the actual frequency offset compensation capability is dependent on the toggle rate of the input data and the input jitter. Only applicable configuration values are listed.

***Table 11 •*** **SERDES1G Loop Bandwidth**

| DES_BW_ANA | Limits |
|---|---|
| 4 | 1953 ppm |
| 5 | 977 ppm |
| 6 | 488 ppm |
| 7 | 244 ppm |

## 4.2.4 SERDES1G Serializer Configuration

The serializer provides the ability to align the phase of the internal clock and data to a selected source (SERDES1G_SER_CFG.SER_ENALI). The phase align logic is used when SERDES1G operates in the facility loopback mode.

### 4.2.5 SERDES1G Input Buffer Configuration

The SERDES1G input buffer supports configuration options for:

- 100BASE-FX mode support
- Signal detection, threshold configurable
- Configurable equalization including corner frequency configuration for the equalization filter
- DC voltage offset compensation
- Configurable common-mode voltage (CMV) termination
- Selectable hysteresis, configurable hysteresis levels

When the SerDes interface operates in 100BASE-FX mode, the input buffer of the SERDES1G macro must also be configured for 100BASE-FX (SERDES1G_IB_CFG.IB_FX100_ENA).

The input buffer provides an option to configure the threshold level of the signal detect circuit to adapt to different input amplitudes. The signal detect circuit can be configured by SERDES1G_IB_CFG.IB_ENA_DETLEV and SERDES1G_IB_CFG.IB_DET_LEV.

The SERDES1G block offers options to compensate for channel loss. Degraded signals can be equalized, and the corner frequency of the equalization filter can be adapted to the channel behavior. The equalization settings are configured by SERDES1G_IB_CFG.IB_EQ_GAIN and SERDES1G_IB_CFG.IB_CORNER_FREQ.

The SERDES1G block compensates for possible DC-offset that can distort the received input signal by enabling SERDES1G_IB_CFG.IB_ENA_OFFSET_COMP during normal reception.

The common-mode voltage (CMV) input termination can be set to either an internal reference voltage or to $V_{DD_A}$. To allow external DC-coupling of the input buffer to an output buffer, set the CMV input termination to the internal reference voltage, with internal DC-coupling disabled. SERDES1G_IB_CFG.IB_ENA_DC_COUPLING controls internal DC-coupling, and SERDES1G_IB_CFG.IB_ENA_CMV_TERM controls CMV input termination. The following modes are defined by CMV input termination and DC-coupling:

- SGMII compliant mode with external AC coupling (IB_ENA_DC_COUPLING =0, IB_ENA_CMV_TERM = 1)
- Vitesse-mode with external DC-coupling to another Vitesse output buffer, which can operate DC-coupled to the input buffer (IB_ENA_DC_COUPLING = 0, IB_ENA_CMV_TERM = 0)
- 100BASE-FX low frequency mode (IB_ENA_DC_COUPLING = 1, IB_ENA_CMV_TERM = 1)

The SERDES1G macro supports input hysteresis, which is required for some standards (SGMII). The hysteresis function is enabled by SERDES1G_IB_CFG.IB_ENA_HYST, and hysteresis levels are defined by SERDES1G_IB_CFG.IB_HYST_LEV.

**Note** Hysteresis and DC offset compensation cannot be enabled at the same time. For more information, see Table 504, page 397.

### 4.2.6 SERDES1G Output Buffer Configuration

The SERDEDS1G output buffer supports configuration options for:

- Configurable amplitude settings
- Configurable slew rate control
- 3 dB de-emphasis selectable
- Idle mode

The output amplitude of the output buffer is controlled by SERDES1G_OB_CFG.OB_AMP_CTRL. It can be adjusted in 50 mV steps from 0.4 V to 1.1 V peak-to-peak differential. The output amplitude also depends on the output buffer's supply voltage. For more information about dependencies between the maximum achievable output amplitude and the output buffer's supply voltage, see Table 818, page 609.

The slew rate is adjustable using SERDES1G_OB_CFG.OB_SLP.

The output buffer supports a fixed 3 dB de-emphasis (SERDES1G_SER_CFG.SER_DEEMPH).

The output buffer supports an idle mode (SERDES1G_SER_CFG.SER_IDLE), which results in an differential peak-to-peak output swing of less than 30 mV.

### 4.2.7 SERDES1G Clock and Data Recovery (CDR) in 100BASE-FX

To enable clock and data recovery when operating SERDES1G in 100BASE-FX mode, set the following register fields:

- SERDES1G_MISC_CFG.DES_100FX_CPMD_ENA = 1
- SERDES1G_IB_CFG.IB_FX100_ENA = 1
- SERDES1G_DES_CFG.DES_CPMD_SEL = 2

### 4.2.8 SERDES1G Energy Efficient Ethernet

The SERDES1G supports Energy Efficient Ethernet as defined in IEEE 802.3az. To enable the low power modes, SERDES1G_MISC_CFG.TX_LPI_MODE_ENA and SERDES1G_MISC_CFG.RX_LPI_MODE_ENA must be set. At this point, the attached PCS takes full control over the high-speed output and input buffer activity.

### 4.2.9 SERDES1G Data Inversion

The data streams in the transmit and the receive direction can be inverted using SERDES1G_MISC_CFG.TX_DATA_INV_ENA and SERDES1G_MISC_CFG.RX_DATA_INV_ENA. This effectively allows for swapping the P and N lines of the high-speed serial link.

## 4.3 SERDES6G

The SERDES6G is a high-speed SerDes interface that operates at 100 Mbps (100BASE-FX), 1 Gbps (SGMII/SerDes), and 4 Gbps (QSGMII). The 100BASE-FX mode is supported by oversampling.

The following table lists the registers associated with SERDES6G.

***Table 12 •*** **SERDES6G Registers**

| Registers | Description | Replication |
|---|---|---|
| SERDES6G_COMMON_CFG | Common configuration | Per SerDes |
| SERDES6G_DES_CFG | Deserializer configuration | Per SerDes |
| SERDES6G_IB_CFG | Input buffer configuration | Per SerDes |
| SERDES6G_IB_CFG1 | Input buffer configuration | Per SerDes |
| SERDES6G_SER_CFG | Serializer configuration | Per SerDes |
| SERDES6G_OB_CFG | Output buffer configuration | Per SerDes |
| SERDES6G_OB_CFG1 | Output buffer configuration | Per SerDes |
| SERDES6G_PLL_CFG | PLL configuration | Per SerDes |
| SERDES6G_MISC_CFG | Miscellaneous configuration | Per SerDes |

For increased performance in specific application environments, SERDES6G supports the following:

- Baud rate support, configurable from 1 Gbps to 4 G, for quarter, half, and full rate modes
- Programmable loop bandwidth and phase regulation for the deserializer
- Configurable input buffer features such as signal detect/loss of signal (LOS) options
- Configurable output buffer features, such as programmable de-emphasis, amplitude drive levels, and slew rate control
- Synchronous Ethernet support
- Loopbacks for system test

### 4.3.1 SERDES6G Basic Configuration

The SERDES6G is enabled in SERDES6G_COMMON_CFG.ENA_LANE. By default, the SERDES6G is held in reset and must be released before the interface is active. This is done through SERDES6G_COMMON_CFG.SYS_RST and SERDES6G_MISC_CFG.LANE_RST.

#### 4.3.1.1 SERDES6G Parallel Interface Configuration

The SERDES6 block includes a parallel data interface, which can operate in two different modes. It must be set according to the mode of operation (SERDES6G_COMMON_CFG.IF_MODE). For 100 Mbps and 1 Gbps operation, the 10-bit mode is used, and for 4 Gbps operation (QSGMII), the 20-bit mode is used.

#### 4.3.1.2 SERDES6G PLL Frequency Configuration

To operate the SERDES6G block at the correct frequency, configure the internal macro as follows. The PLL calibration is enabled through SERDES6G_PLL_CFG.PLL_FSM_ENA.

1. Configure SERDES6G_PLL_CFG.PLL_FSM_CTRL_DATA in accordance with data rates listed in the following two tables.
2. Set SYS_RST = 0 (active) and PLL_FSM_ENA = 0 (inactive).
3. Set SYS_RST = 1 (deactive) and PLL_FSM_ENA = 1 (active).

***Table 13 •*** **PLL Configuration**

| Mode | SERDES6G_PLL_CFG.PLL_FSM_CTRL_DATA |
|---|---|
| SGMII/SerDes, 1 Gbps data | 60 |
| QSGMII, 4 Gbps data | 120 |

#### 4.3.1.3 SERDES6G Frequency Configuration

The following table lists the range of data rates that are supported by SERDES6G.

***Table 14 •*** **SERDES6 Frequency Configuration Registers**

| Configuration | SGMII/SerDes 1 Gbps | QSGMII 4 Gbps |
|---|---|---|
| SERDES6G_PLL_CFG.PLL_ROT_DIR | 1 | 0 |
| SERDES6G_COMMON_CFG.QRATE | 1 | 0 |
| SERDES6G_COMMON_CFG.HRATE | 0 | 0 |

### 4.3.2 SERDES6G Loopback Modes

The SERDES6G interface supports two different loopback modes for testing and debugging data paths: equipment loopback and facility loopback.

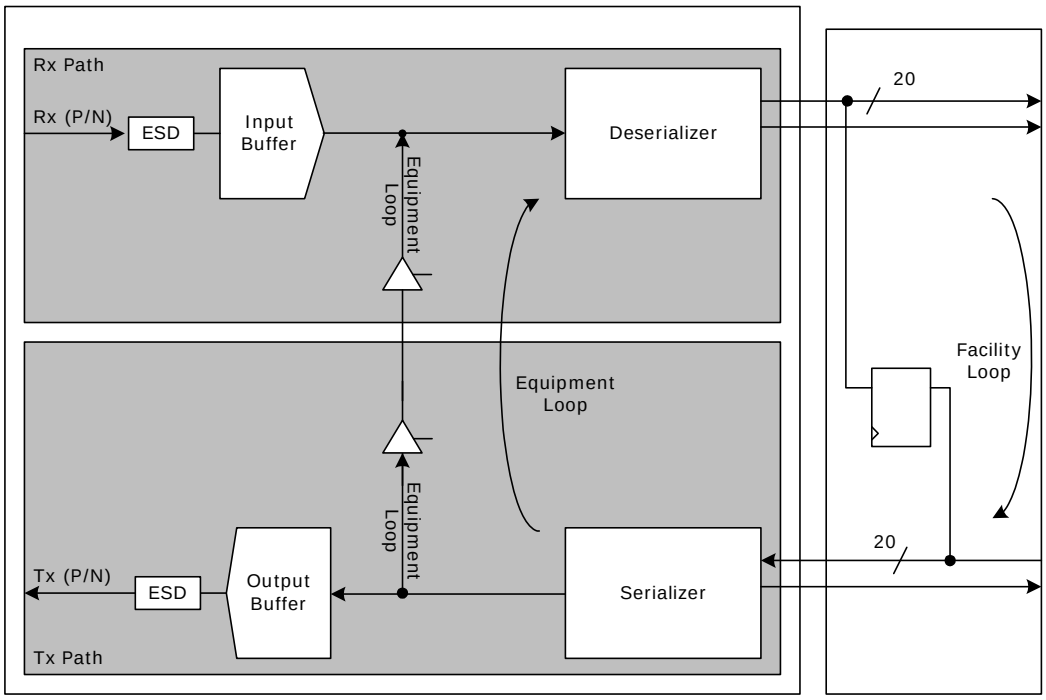
**Equipment loopback (SERDES6G_COMMON_CFG.ENA_ELOOP)** Data is looped back from serializer output to the deserializer input, and the receive clock is recovered. The equipment loopback includes all transmit and receive functions, except for the input and output buffers. The Tx data can still be observed on the output.

**Facility loopback (SERDES6G_COMMON_CFG.ENA_FLOOP)** The clock and parallel data output from deserializer are looped back to the serializer interface. Incoming serial data passes through the input buffer, the CDR, the deserializer, back to the serializer, and finally out through the output buffer.

Only one of the loopbacks can be enabled at the same time.

The following illustration shows the loopback paths for the SERDESG6.

*Figure 7 •* **SERDES Loopback**




## 4.3.3 SERDES6G Deserializer Configuration

The SERDES6G block includes digital control logic that interacts with the analog modules within the block and compensates for the frequency offset between the received data and the internal high-speed reference clock. To gain high jitter performance, the phase regulation is a PI-type regulator, whose proportional (P) and integrative (I) characteristics can be independently configured.

The integrative part of the phase regulation loop is configured in SERDES6G_DES_CFG.DES_PHS_CTRL. The limits of the integrator are programmable, allowing different settings for the integrative regulation while guaranteeing that the proportional part still is stronger than the integrative part. Integrative regulation compensates frequency modulation from DC up to cut-off frequency. Frequencies above the cut-off frequency are compensated by the proportional part.

The DES_BW_HYST register field controls the time constant of the integrator independently of the proportional regulator. The range of DES_BW_HYST is programmable as follows:

- Full rate mode = 3 to 7
- Quarter-rate mode = 1 to 7

The lower the configuration setting, the smaller the time-constant of the integrative regulation. For normal operation, configure DES_BW_HYST to 5.

The cut-off-frequency is calculated to:

fco = 1/(2 × PI × 128 × PLL period × 32 × 2^(DES_BW_HYST + 1 – DES_BW_ANA))

PLL period = 1/(*n* × data rate)

where, *n* = 1 (full rate mode) or 4 (quarter-rate mode)

The integrative regulator can compensate a static frequency offset within the programmed limits down to a remaining frequency error of below 4 ppm. In steady state, the integrator toggles between two values around the exact value, and the proportional part of the phase regulation takes care of the remaining phase error.

After a device reset, the phase regulation may be 180° out of phase compared to the incoming data, resulting in a deadlock condition at the sampling stage of the deserializer. To prevent this situation, the SERDES6G provides a 180° deadlock protection mechanism

(SERDES6G_DES_CFG.DES_MBTR_CTRL). If the deadlock protection mechanism is enabled, a small frequency offset is applied to the phase regulation loop. The offset is sufficient to move the sampling point out of the 180° deadlock region, while at the same time, small enough to allow the regulation loop to compensate when the sample point is within the data eye.

The loop bandwidth for the proportional part of the phase regulation loop is controlled by configuring SERDES6G_DES_CFG.DES_BW_ANA.

The fastest loop bandwidth setting (lowest configuration value) results in a loop bandwidth that is equal to the maximum frequency offset compensation capability. For improved jitter performance, use a setting with sufficient margin to track the expected frequency offset rather than using the maximum frequency offset. For example, if a 100 ppm offset is expected, use a setting that is four times higher than the offset. For more information about possible bandwidth selections, see Table 503, page 396 and Table 516, page 404.

The following table provides the limits for the frequency offset compensation. The values are theoretical limits for input signals without jitter, because the actual frequency offset compensation capability is dependent on the toggle rate of the input data and the input jitter. Note that only applicable configuration values are listed. HRATE and QRATE are the configuration settings of SERDES6G_COMMON_CFG.HRATE and SERDES6G_COMMON_CFG.QRATE.

***Table 15 •*** **SERDES6G Loop Bandwidth**

| DES_BW_ANA | Limits when HRATE = 0 QRATE = 0 | Limits when HRATE = 1 QRATE = 0 | Limits when HRATE = 0 QRATE = 1 |
|---|---|---|---|
| 2 | | | 1953 ppm |
| 3 | | 1953 ppm | 977 ppm |
| 4 | 1953 ppm | 977 ppm | 488 ppm |
| 5 | 977 ppm | 488 ppm | 244 ppm |
| 6 | 488 ppm | 244 ppm | 122 ppm |
| 7 | 244 ppm | 122 ppm | 61 ppm |

### 4.3.4 SERDES6G Serializer Configuration

The serializer provides the ability to align the phase of the internal clock and data to a selected source (SERDES6G_SER_CFG.SER_ENALI). The phase align logic is used when SERDES6G operates in the facility loopback mode.

### 4.3.5 SERDES6G Input Buffer Configuration

The SERDES6G input buffer supports configuration options for:

- Automatic input voltage offset compensation
- Loss of signal detection

The input buffer is normally AC-coupled and therefore the common-mode termination is switched off (SERDES6G_IB_CFG1.IB_CTERM_ENA). In order to support type-2 loads (DC-coupling at 1.0 V termination voltage) according to the OIF CEI specifications, common-mode termination must be enabled.

The sensitivity of the level detect circuit can be adapted to the input signal's characteristics (amplitude and noise). The threshold value for the level detect circuit is set in SERDES6G_IB_CFG.IB_VBCOM. The default value is suitable for normal operation.

When the SerDes interface operates in 100BASE-FX mode, the input buffer of the SERDES6G macro must also be configured for 100BASE-FX (SERDES6G_IB_CFG.IB_FX100_ENA).

During test or reception of low data rate signals (for example, 100BASE-FX), the DC-offset compensation must be disabled. For all other modes, the DC-offset compensation must be enabled for

optimized performance. DC-offset compensation is controlled by SERDES6G_IB_CFG1.IB_ENA_OFFSAC and SERDES6G_IB_CFG1.IB_ENA_OFFSDC.

### 4.3.6 SERDES6G Output Buffer Configuration

The SERDEDS6G output buffer supports the following configuration options:

- Amplitude control
- De-emphasis and output polarity inversion
- Slew rate control
- Skew adjustment
- Idle mode

The maximum output amplitude of the output buffer depends on the output buffer's supply voltage. For interface standards requiring higher output amplitudes (backplane application or interface to optical modules, for example), the output buffer can be supplied from a 1.2 V instead of a 1.0 V supply. By default, the output buffer is configured for 1.2 V mode, because enabling the 1.0 V mode when supplied from 1.2 V must be avoided. The supply mode is configured by SERDES6G_OB_CFG.OB_ENA1V_MODE.

The output buffer supports a four-tap pre-emphasis realized by one pre-cursor, the center tap, and two post cursors. The pre-cursor coefficient, C0, is configured by SERDES6G_SER_CFG.OB_PREC. C0 is a 5-bit value, with the most significant bit defining the polarity. The lower 4-bit value is hereby defined as B0. The first post-cursor coefficient, C2, is configured by SERDES6G_OB_CFG.OB_POST0. C2 is a 6-bit value, with the most significant bit defining the polarity. The lower 5-bit value is hereby defined as B2. The second post-cursor coefficient, C3, is configured by SERDES6G_SER_CFG.OB_POST1. C3 is 5-bit value, with the most significant bit defining the polarity. The lower 4-bit value is hereby defined as B3. The center-tap coefficient, C1, is a 6-bit value. Its polarity can be programmed by SERDES6G_OB_CFG.OB_POL, which is defined as p1. For normal operation SERDES6G_OB_CFG.OB_POL must be set to 1. The value of the 6 bits forming C1 is calculated by the following equation.

**Equation 1:** C1: (64 – (B0 + B2 + B3)) × p1

The output amplitude is programmed by SERDES6G_OB_CFG1.OB_LEV, which is a 6-bit value. This value is internally increased by 64 and defines the amplitude coefficient K. The range of K is therefore 64 to 127. The differential peak-peak output swing is given by 8.75 mV × K. The maximum peak-peak output swing depends on the data stream and can be calculated to:

**Equation 2:** $H(Z) = 4.375\ mVpp \times K \times (C0 \times z^{1} + C1 \times z^{0} + C2 \times z^{-1} + C3 \times z^{-2})/64$

with $z^{n}$ denoting the current bits of the data pattern defining the amplitude of Z. The output amplitude also depends on the output buffer's supply voltage. For more information about the dependencies between the maximum achievable output amplitude and the output buffer's supply voltage, see Table 816, page 607.

The configuration bits are summarized in the following table.

***Table 16 •*** **De-Emphasis and Amplitude Configuration**

| Configuration | Value | Description |
|---|---|---|
| OB_PREC | Signed 5-bit value | Pre-cursor setting C0<br>Range is –15 to 15 |
| OB_POST0 | Signed 6-bit value | First post-cursor setting C2<br>Range is –31 to 31 |
| OB_POST1 | Signed 5-bit value | Second post-cursor setting C3<br>Range is –15 to 15 |
| OB_LEV | Unsigned 6-bit value | Amplitude coefficient, K = OB_LEV + 64<br>Range is 0 to 63 |
| OB_POL | 0<br>1 | Non-inverting mode<br>Inverting mode |

The output buffer provides additional options to configure its behavior. These options are:

- Idle mode:
  Enabling idle mode (SERDES6G_OB_CFG.OB_IDLE) results in a remaining voltage of less than 30 mV at the buffers differential outputs.
- Slew Rate:
  Slew rate can be controlled by two configuration settings. SERDES6G_OB_CFG.OB_SR_H provides coarse adjustments whereas SERDES6G_OB_CFG.OB_SR provides fine adjustments.
- Skew control:
  In 1 Gbps SGMII mode, skew adjustment is controlled by SERDES6G_OB_CFG1.OB_ENA_CAS. Skew control is not applicable to other modes.

### 4.3.7 SERDES6G Clock and Data Recovery (CDR) in 100BASE-FX

To enable clock and data recovery when operating SERDES6G in 100BASE-FX mode, set the following register fields:

- SERDES6G_MISC_CFG.DES_100FX_CPMD_ENA = 1
- SERDES6G_IB_CFG.IB_FX100_ENA = 1
- SERDES6G_DES_CFG.DES_CPMD_SEL = 2

### 4.3.8 SERDES6G Energy Efficient Ethernet

The SERDES6G block supports Energy Efficient Ethernet as defined in IEEE 802.3az. To enable the low power modes, set SERDES6G_MISC_CFG.TX_LPI_MODE_ENA and SERDES6G_MISC_CFG.RX_LPI_MODE_ENA. At this point, the attached PCS takes full control over the high-speed output and input buffer activity.

### 4.3.9 SERDES6G Data Inversion

The data streams in the transmit and the receive direction can be inverted using SERDES6G_MISC_CFG.TX_DATA_INV_ENA and SERDES6G_MISC_CFG.RX_DATA_INV_ENA. This effectively allows for swapping the P and N lines of the high-speed serial link.

### 4.3.10 SERDES6G Signal Detection Enhancements

Signal detect information from the SERDES6G macro is normally directly passed to the attached PCS. It is possible to enable a hysteresis such that the signal detect condition must be active or inactive for a certain time before it is signaled to the attached PCS.

The signal detect assertion time (the time signal detect must be active before the information is passed to a PCS) is programmable in SERDES6G_DIG_CFG.SIGDET_AST. The signal detect de-assertion time (the time signal detect must be inactive before the information is passed to a PCS) is programmable in SERDES6G_DIG_CFG.SIGDET_DST.

### 4.3.11 SERDES6G High-Speed I/O Configuration Bus

The high-speed SerDes macros are configured using the high-speed I/O configuration bus (MCB), which is a serial bus connecting the configuration register set with all the SerDes macros. The HSIO::MCB_SERDES1G_ADDR_CFG register is used for SERDES1G macros and HSIO::MCB_SERDES6G_ADDR_CFG register is used for SERDES6G macros. The configuration busses are used for both writing to and reading from the macros.

The SERDES6G macros are programmed as follows:

- Program the configuration registers for the SERDES6G macro. For more information about configuration options, see SERDES6G, page 26.
- Transfer the configuration from the configuration registers to one or more SerDes macros by writing the address of the macro (MCB_SERDES6G_ADDR_CFG.SERDES6G_ADDR) and initiating the write access (MCB_SERDES6G_ADDR_CFG.SERDES6G_WR_ONE_SHOT).
- The SerDes macro address is a mask with one bit per macro so that one or more macros can be programmed at the same time.
- The MCB_SERDES6G_ADDR_CFG.SERDES6G_WR_ONE_SHOT are automatically cleared when the writing is done.

The configuration and status information in the SERDES6G macros can be read as follows:

- Transfer the configuration and status from one or more SerDes macros to the configuration registers by writing the address of the macro (MCB_SERDES6G_ADDR_CFG.SERDES6G_ADDR) and initiating the read access (MCB_SERDES6G_ADDR_CFG.SERDES6G_RD_ONE_SHOT).
- The SerDes macro address is a mask with one bit per macro so that configuration and status information from one or more macros can be read at the same time. When reading from more than one macro, the results from each macro are OR'ed together.
- The MCB_SERDES6G_ADDR_CFG.SERDES6G_RD_ONE_SHOT are automatically cleared when the reading is done.

The SERDES1G macros are programmed similarly to the SERDES6G macros, except that MCB_SERDES1G_ADDR_CFG must be used for register access. For more information about configuration options, see SERDES1G, page 22.

## 4.4 Copper Transceivers

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices include low-power Gigabit Ethernet transceivers. The devices include the following number of transceivers:

- VSC7424-02 includes 8 transceivers, numbered 0 through 7
- VSC7425-02, VSC7426-02, and VSC7427-02 include 12 transceivers, numbered 0 through 11

This section describes the high-level functionality and operation of the built-in transceivers. The integration is kept as close to multi-chip PHY and switch designs as possible. This allows a fast path for software already running in a similar distributed design while still benefiting from the cost savings provided by the integration.

### 4.4.1 Register Access

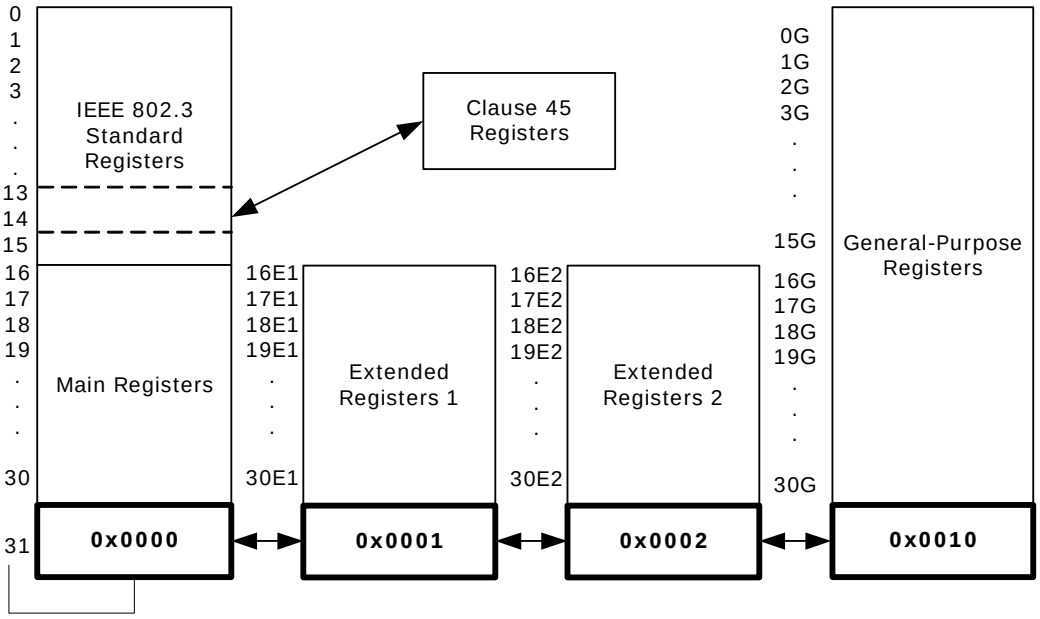
The registers of the integrated transceivers are not placed in the memory map of the switch, but are attached instead to the built-in MII management controller 0 of the devices. As a result, PHY registers are accessed indirectly through the switch registers. For more information, see MII Management Controller, page 174.

In addition to providing the IEEE 802.3 specified 16 MII Standard Set registers, the PHYs contain an extended set of registers that provide additional functionality. The devices support the following types of registers:

- IEEE Clause 22 device registers with addresses from 0 to 31
- Two pages of extended registers with addresses from 16E1 through 30E1 and 16E2 through 30E2
- General-purpose registers with addresses from 0G to 30G
- IEEE Clause 45 devices registers accessible through the Clause 22 registers 13 and 14 to support IEEE 802.3az Energy Efficient Ethernet registers

The memory mapping is controlled through PHY_MEMORY_PAGE_ACCESS::PAGE_ACCESS_CFG. The following illustration shows the relationship between the device registers and their address spaces.

*Figure 8 •* **Register Space Layout**




#### 4.4.1.1 Broadcast Write

The PHYs can be configured to accept MII PHY register write operations regardless of the destination address of these writes. This is enabled in PHY_CTRL_STAT_EXT::BROADCAST_WRITE_ENA. This enabling allows similar configurations to be sent quickly to multiple PHYs without having to do repeated MII PHY write operations. This feature applies only to writes; MII PHY register read operations are still interpreted with “correct” address.

#### 4.4.1.2 Register Reset

The PHY can be reset through software. This is enabled in PHY_CTRL::SOFTWARE_RESET_ENA. Enabling this field initiates a software reset of the PHY. Fields that are not described as sticky are returned to their default values. Fields that are described as sticky are only returned to defaults if sticky-reset is disabled through PHY_CTRL_STAT_EXT::STICKY_RESET_ENA. Otherwise, they retain their values from prior to the software reset. A hardware reset always brings all PHY registers back to their default values.

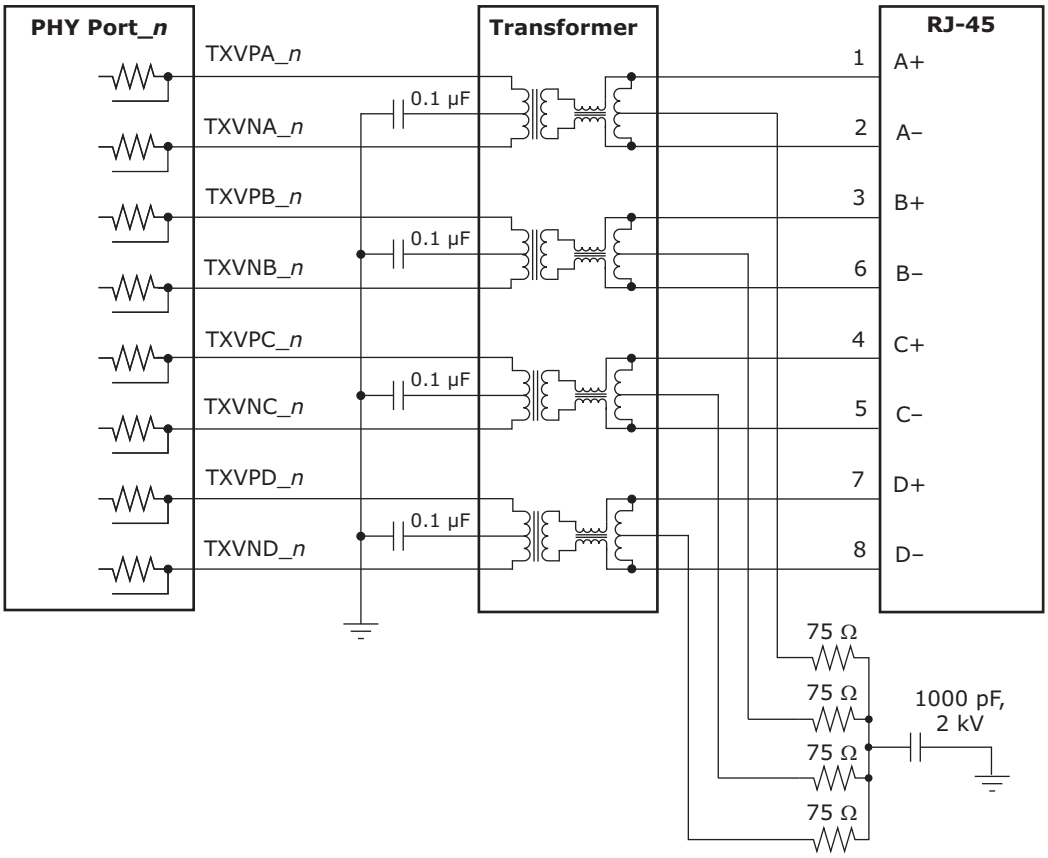
### 4.4.2 Cat5 Twisted Pair Media Interface

The twisted pair interfaces are compliant with IEEE 802.3-2008 and IEEE 802.3az for Energy Efficient Ethernet.

#### 4.4.2.1 Voltage-Mode Line Driver

Unlike many other gigabit PHYs, this PHY uses a patented voltage-mode line driver that allows it to fully integrate the series termination resistors (required to connect the PHY’s Cat5 interface to an external 1:1 transformer). Also, the interface does not require placement of an external voltage on the center tap of the magnetic. The following illustration shows the connections.

*Figure 9 •* **Cat5 Media Interface**




### 4.4.2.2 Cat5 Autonegotiation and Parallel Detection

The integrated transceivers support twisted pair autonegotiation as defined by clause 28 of the IEEE 802.3-2008. The autonegotiation process evaluates the advertised capabilities of the local PHY and its link partner to determine the best possible operating mode. In particular, auto-negotiation can determine speed, duplex configuration, and master or slave operating modes for 1000BASE-TX. Auto-negotiation also allow the devices to communicate with the link partner (through the optional “next pages”) to set attributes that may not otherwise be defined by the IEEE standard.

If the Cat5 link partner does not support auto negotiation, the devices automatically use parallel detection to select the appropriate link speed.

Auto-negotiation can be disabled by clearing PHY_CTRL.AUTONEG_ENA. If auto-negotiation is disabled, the state of the SPEED_SEL_MSB_CFG, SPEED_SEL_LSB_CFG, and DUPLEX_MODE_CFG fields in the PHY_CTRL register determine the device operating speed and duplex mode. Note that while 10BASE-T and 100BASE-T do not require auto-negotiation, clause 40 defines that 1000BASE-T require auto-negotiation.

### 4.4.2.3 1000BASE-T Forced Mode Support

The integrated transceivers provides support for a 1000BASE-T forced test mode. In this mode, the PHY can be forced into 1000BASE-T mode and does not require manual setting of master/slave at the two ends of the link. This mode is only for test purposes. Do not use in normal operation. To configure a PHY in this mode, set PHY_EEE_CTRL.FORCE_1000BT_ENA = 1, with PHY_CTRL.SPEED_SEL_LSB_CFG = 1 and PHY_CTRL.SPEED_SEL_LSB_CFG = 0.

### 4.4.2.4 Automatic Crossover and Polarity Detection

For trouble-free configuration and management of Ethernet links, the integrated transceivers include a robust automatic crossover detection feature for all three speeds on the twisted-pair interface (10BASE-

T, 100BASE-T, and 1000BASE T). Known as HP Auto-MDIX, the function is fully compliant with clause 40 of the IEEE 802.3-2002.

Additionally, the devices detect and correct polarity errors on all MDI pairs—a useful capability that exceeds the requirements of the standard.

Both HP Auto-MDIX detection and polarity correction are enabled in the device by default. You can change the default settings using fields POL_INV_DIS and PAIR_SWAP_DIS in the PHY_BYPASS_CTRL register. Status bits for each of these functions are located in register PHY_AUX_CTRL_STAT.

The integrated transceivers can be configured to perform HP Auto-MDIX, even when auto-negotiation is disabled (PHY_CTRL.AUTONEG_ENA = 0) and the link is forced into 10/100 speeds. To enable the HP Auto-MDIX feature, set PHY_BYPASS_CTRL.FORCED_SPEED_AUTO_MDIX_DIS to 0.

The HP Auto-MDIX algorithm successfully detects, corrects, and operates with any of the MDI wiring pair combinations listed in the following table.

***Table 17 •*** **Supported MDI Pair Combinations**

| RJ-45 Pin Pairings | | | | |
|---|---|---|---|---|
| **1, 2** | **3, 6** | **4, 5** | **7, 8** | **Mode** |
| A | B | C | D | Normal MDI |
| B | A | D | C | Normal MDI-X |
| A | B | D | C | Normal MDI with pair swap on C and D pair |
| B | A | C | D | Normal MDI-X with pair swap on C and D pair |

#### 4.4.2.5 Manual MDI/MDI-X Setting

As an alternative to HP Auto-MDIX detection, the PHY can be forced to be MDI or MDI-X using PHY_EXT_MODE_CTRL.FORCE_MDI_CROSSOVER_ENA. Setting this field to 10 forces MDI, and setting 11 forces MDI-X. Leaving the bits 00 enables the MDI/MDI-X setting to be based on FORCED_SPEED_AUTO_MDIX_DIS and PAIR_SWAP_DIS in the register PHY_BYPASS_CTRL.

#### 4.4.2.6 Link Speed Downshift

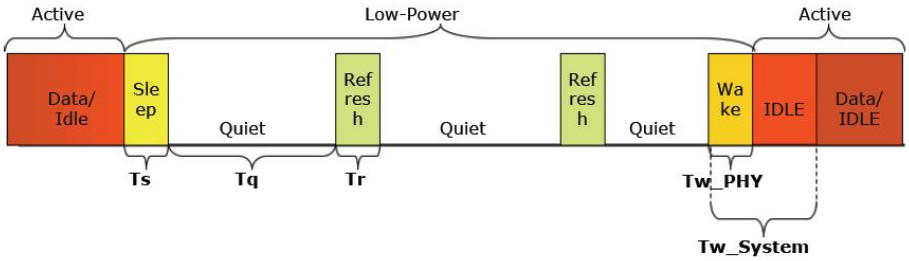
For operation in cabling environments that are incompatible with 1000BASE-T, the devices provide an automatic link speed “downshift” option. When enabled, the devices automatically change their 1000BASE-T auto-negotiation advertisement to the next slower speed after a set number of failed attempts at 1000BASE-T. No reset is required to exit this state if a subsequent link partner with 1000BASE-T support is connected. This is useful in setting up in networks using older cable installations that may include only pairs A and B and not pairs C and D.

Link speed downshifting is configured and monitored using SPEED_DOWNSHIFT_STAT, SPEED_DOWNSHIFT_CFG, and SPEED_DOWNSHIFT_ENA in the register PHY_CTRL_EXT3.

#### 4.4.2.7 Energy Efficient Ethernet

The integrated transceivers support IEEE 802.3az Energy Efficient Ethernet (EEE) currently in development. This new standard provides a method for reducing power consumption on an Ethernet link during times of low use. It uses Low Power Idles (LPI) to achieve this objective.

***Figure 10 •*** **Energy Efficient Ethernet**




| Term | Description |
|---|---|
| **Sleep Time (Ts)** | **Duration PHY sends Sleep symbols before going Quiet.** |
| **Quiet Duration (Tq)** | **Duration PHY remains Quiet before it must wake for Refresh period.** |
| **Refresh Duration (Tr)** | **Duration PHY sends Refresh symbols for timing recovery and coefficient synchronization.** |
| **PHY Wake Time (Tw_PHY)** | **Duration PHY takes to resume to Active state after decision to Wake.** |
| **System Wake Time (Tw_System)** | **Wait period where no data is transmitted to give the receiving system time to wake up.** |

Using LPI, the usage model for the link is to transmit data as fast as possible and then return to a low power idle state. Energy is saved on the link by cycling between active and low power idle states. Power is reduced during LPI by turning off unused circuits and, using this method, energy use scales with bandwidth utilization.

The transceivers use LPI to optimize power dissipation in 100BASE-TX and 1000BASE-T operation. In addition, IEEE 802.3az defines a 10BASE-Te mode that reduces transmit signal amplitude from 5 V to approximately 3.3 V, peak-to-peak. This mode reduces power consumption in 10 Mbps link speed and can fully interoperate with legacy 10BASE-T compliant PHYs over 100 m Cat5 cable or better.

To configure the transceivers in 10BASE-Te mode, set PHY_EEE_CTRL.EEE_LPI_RX_100BTX_DIS to 1 for each port. Additional Energy Efficient Ethernet features are controlled through Clause 45 registers as defined in Clause 45 registers to Support Energy Efficient Ethernet.

## 4.4.3 LED Interface

The devices output two LED signals per port, LED0 and LED1, through direct-drive signal outputs. The polarity of the LED outputs is programmable and can be changed through PHY_EEE_CTRL.INV_LED_POL_ENA. The default polarity is active low.

The devices also have a serial LED interface if more than two LEDs per port are required. For more information, see Serial GPIO Controller, page 178.

### 4.4.3.1 LED Modes

Each direct-drive LED pin can be configured to display different status information that can be selected by setting the LED mode in register PHY_LED_MODE_SEL. The modes listed in the following table are equivalent to the setting used in PHY_LED_MODE_SEL to configure each LED pin. The default LED state is active low and can be changed by modifying the value in PHY_EEE_CTRL.INV_LED_POL_ENA. The blink/pulse-stretch is dependent on the LED behavior settings in PHY_LED_BEHAVIOR_CTRL.

***Table 18 •*** **LED Modes**

| Mode | Function Name | LED State and Description |
|---|---|---|
| 0 | Link/Activity | 1: No link in any speed on any media interface.<br>0: Valid link at any speed on any media interface.<br>Blink or pulse-stretch: Valid link at any speed on any media interface with activity present. |

*Table 18 •* **LED Modes *(continued)***

| Mode | Function Name | LED State and Description |
|---|---|---|
| 1 | Link1000/Activity | 1: No link in 1000BASE-T or 1000BASE-X.<br>0: Valid 1000BASE-T or 1000BASE-X.<br>Blink or pulse-stretch: Valid 1000BASE-T or 1000BASE-X link with activity present. |
| 2 | Link100/Activity | 1: No link in 100BASE-TX or 100BASE-FX.<br>0: Valid 100BASE-TX or 100BASE-FX.<br>Blink or pulse-stretch: Valid 100BASE-TX or 100BASE-FX link with activity present. |
| 3 | Link10/Activity | 1: No link in 10BASE-T.<br>0: Valid 10BASE-T link.<br>Blink or pulse-stretch: Valid 10BASE-T link with activity present. |
| 4 | Link100/1000/Activity | 1: No link in 100BASE-TX, 100BASE-FX, 1000BASE-X, or 1000BASE-T.<br>0: Valid 100BASE-TX, 100BASE-FX, 1000BASE-X, or 1000BASE-T link.<br>Blink or pulse-stretch: Valid 100BASE-TX, 100BASE-FX, 1000BASE-X, or 1000BASE-T link with activity present. |
| 5 | Link10/1000/Activity | 1: No link in 10BASE-T, 1000BASE-X, or 1000BASE-T.<br>0: Valid 10BASE-T, 1000BASE-X, or 1000BASET-T link.<br>Blink or pulse-stretch: Valid 10BASE-T, 1000BASE-X, or 1000BASE-T link with activity present. |
| 6 | Link10/100/Activity | 1: No link in 10BASE-T, 100BASE-FX, or 100BASE-TX.<br>0: Valid 10BASE-T 100BASE-FX, or 100BASE-TX, link.<br>Blink or pulse-stretch: Valid 10BASE-T, 100BASE-FX, or 100BASE-TX link with activity present. |
| 7 | Reserved. | Reserved. |
| 8 | Duplex/Collision | 1: Link established in half-duplex mode, or no link established.<br>0: Link established in full-duplex mode.<br>Blink or pulse-stretch: Link established in half-duplex mode but collisions are present. |
| 9 | Collision | 1: No collision detected.<br>Blink or pulse-stretch: Collision detected. |
| 10 | Activity | 1: No activity present.<br>Blink or pulse-stretch: Activity present. |
| 11 | Reserved | Reserved. |
| 12 | Auto-Negotiation Fault | 1: No autonegotiation fault present.<br>0: Autonegotiation fault occurred. |
| 13 | Reserved. | Reserved. |
| 14 | Force LED Off | 1: De-asserts the LED |
| 15 | Force LED On | 0: Asserts the LED |

#### 4.4.3.2 LED Behavior

Several LED behaviors can be programmed into the PHYs. Use the settings in registers PHY_LED_BEHAVIOR_CTRL and PHY_EXT_MODE_CTRL to program the following LED behaviors:

**LED Combine (LEDx_COMBINE_DIS)** Enables an LED to display the status for a combination of primary and secondary modes. This can be enabled or disabled for each LED pin. For example, a copper link running in 1000BASE-T mode and activity present can be displayed with one LED by configuring an

LED pin to Link1000/Activity mode. The LED asserts when linked to a 1000BASE-T partner and also blinks or performs pulse-stretch when activity is either transmitted by the PHY or received by the link partner. When disabled, the LED combine feature only provides status of the selected primary function. In this example, only Link1000 asserts the LED, and the secondary mode, activity, does not display if the combined feature is disabled.

**LED Blink or Pulse-Stretch (LEDx_PULSE_STRETCH_ENA)** This behavior is used for activity and collision indication. This can be uniquely configured for each LED pin. Activity and collision events can occur randomly and intermittently throughout the link-up period. Blink is a 50% duty cycle oscillation of asserting and de-asserting an LED pin. Pulse-stretch guarantees that an LED is asserted and de-asserted for a specific period of time when activity is either present or not present. These rates can also be configured using a register setting.

**Rate of LED Blink or Pulse-Stretch (BLINK_RATE_CFG)** This behavior controls the LED blink rate or pulse-stretch length when blink/pulse-stretch is enabled on an LED pin. The blink rate, which alternates between a high and low voltage level at a 50% duty cycle, can be set to 2.5 Hz, 5 Hz, 10 Hz, or 20 Hz. For pulse-stretch, the rate can be set to 50 ms, 100 ms, 200 ms, or 400 ms. The blink rate selection for PHY0 globally sets the rate used for all LED pins on all PHY ports.

**LED Pulsing Enable (PULSING_ENA)** To provide additional power savings, the LEDs (when asserted) can be pulsed at 5 kHz, 20% duty cycle.

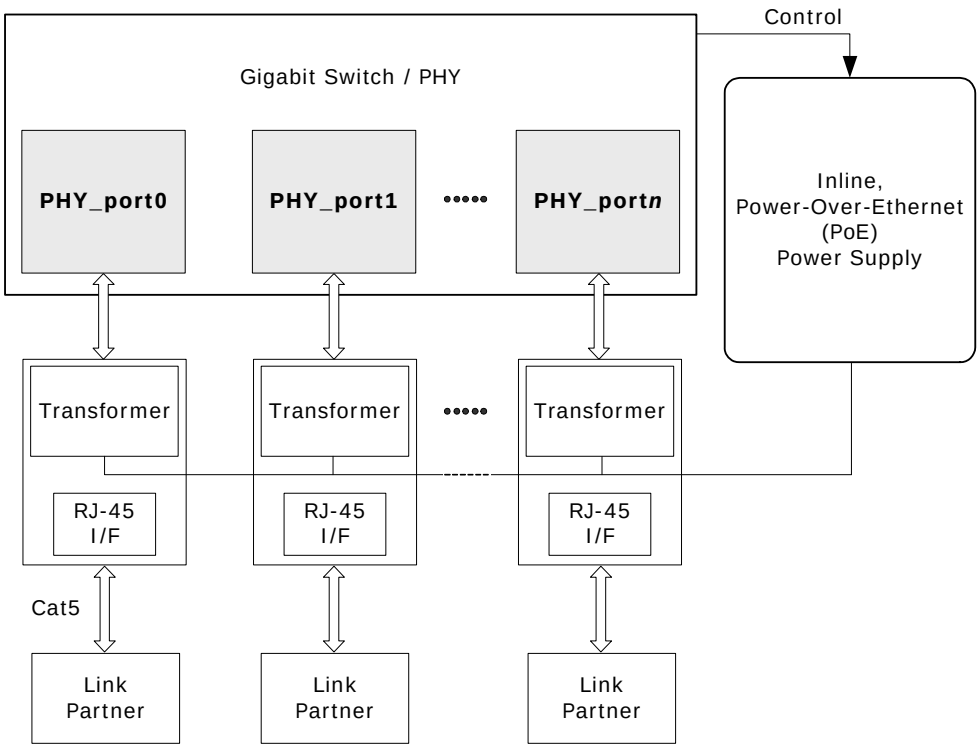
**LED Blink After Reset (LED_BLINK_SUPPRESS)** The LEDs blink for one second after power-up and after any time all resets are de-asserted.

### 4.4.4 Ethernet Inline Powered Devices

The integrated transceivers can detect legacy inline powered devices in Ethernet network applications. The inline powered detection capability can be part of a system that allows for IP-phone and other devices, such as wireless access points, to receive power directly from their Ethernet cable, similar to office digital phones receiving power from a Private Branch Exchange (PBX) office switch over the telephone cabling. This can eliminate the need of an external power supply for an IP-phone. It also enables the inline powered device to remain active during a power outage (assuming the Ethernet switch is connected to an uninterrupted power supply, battery, back-up power generator, or some other uninterruptable power source).

The following illustration shows an example of this type of application.

***Figure 11 •*** **Inline Powered Ethernet Switch**




The following procedure describes the process that an Ethernet switch must perform to process inline power requests made by a link partner (LP); that is, in turn, capable of receiving inline power.

1. Enable the inline powered device detection mode on each transceiver using its serial management interface. Set PHY_CTRL_EXT4.INLINE_POW_DET_ENA to 1.
2. Ensure that the Auto-Negotiation Enable bit (register 0.12) is also set to 1. In the application, the devices send a special Fast Link Pulse (FLP) signal to the LP. Reading PHY_CTRL_EXT4.INLINE_POW_DET_STAT returns 00 during the search for devices that require Power-over-Ethernet (PoE).
3. The transceiver monitors its inputs for the FLP signal looped back by the LP. An LP capable of receiving PoE loops back the FLP pulses when the LP is in a powered-down state. This is reported when PHY_CTRL_EXT4.INLINE_POW_DET_STAT reads back 01. If an LP device does not loop back the FLP after a specific time, PHY_CTRL_EXT4.INLINE_POW_DET_STAT automatically resets to 10.
4. If the transceiver reports that the LP needs PoE, the Ethernet switch must enable inline power on this port, externally of the PHY.
5. The PHY automatically disables inline powered device detection if PHY_CTRL_EXT4.INLINE_POW_DET_STAT automatically resets to 10, and then automatically changes to its normal auto-negotiation process. A link is then auto-negotiated and established when the link status bit is set (PHY_STAT.LINK_STAT is set to 1).
6. In the event of a link failure (indicated when PHY_STAT.LINK_STAT reads 0), the inline power must be disabled to the inline powered device external to the PHY. The transceiver disables its normal auto-negotiation process and re-enables its inline powered device detection mode.

### 4.4.5 IEEE 802.3af PoE Support

The integrated transceivers are also compatible with switch designs intended for use in systems that supply power to Data Terminal Equipment (DTE) by means of the MDI or twisted pair cable, as described in clause 33 of the IEEE 802.3af.

### 4.4.6 ActiPHY™ Power Management

In addition to the IEEE-specified power-down control bit (PHY_CTRL.POWER_DOWN_ENA), the devices also include an ActiPHY power management mode for each PHY. The ActiPHY mode enables

support for power-sensitive applications. It uses a signal detect function that monitors the media interface for the presence of a link to determine when to automatically power-down the PHY. The PHY “wakes up” at a programmable interval and attempts to wake-up the link partner PHY by sending a burst of FLP over copper media.

The ActiPHY power management mode in the integrated transceivers is enabled on a per-port basis during normal operation at any time by setting PHY_AUX_CTRL_STAT.ACTIPHY_ENA to 1.

Three operating states are possible when ActiPHY mode is enabled:

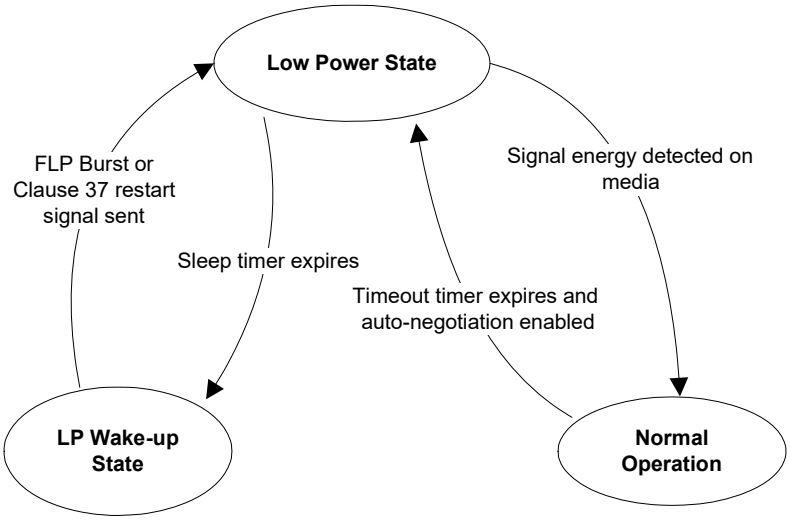
- Low power state
- LP wake-up state
- Normal operating state (link up state)

The PHY switches between the low power state and the LP wake-up state at a programmable rate (the default is two seconds) until signal energy is detected on the media interface pins. When signal energy is detected, the PHY enters the normal operating state. If the PHY is in its normal operating state and the link fails, the PHY returns to the low power state after the expiration of the link status time-out timer. After reset, the PHY enters the low power state.

When auto-negotiation is enabled in the PHY, the ActiPHY state machine operates as described. If auto-negotiation is disabled and the link is forced to use 10BT or 100BTX modes while the PHY is in its low power state, the PHY continues to transition between the low power and LP wake-up states until signal energy is detected on the media pins. At that time, the PHY transitions to the normal operating state and stays in that state even when the link is dropped. If auto-negotiation is disabled while the PHY is in the normal operation state, the PHY stays in that state when the link is dropped and does not transition back to the low power state.

The following illustration shows the relationship between ActiPHY states and timers.

***Figure 12 •*** **ActiPHY State Diagram**




### 4.4.6.1 Low Power State

All major digital blocks are powered down in the lower power state.

In this state, the PHY monitors the media interface pins for signal energy. The PHY comes out of low power state and transitions to the normal operating state when signal energy is detected on the media. This happens when the PHY is connected to one of the following:

- Auto-negotiation capable link partner
- Another PHY in enhanced ActiPHY LP wake-up state

In the absence of signal energy on the media pins, the PHY transitions from the low power state to the LP wake-up state periodically based on the programmable sleep timer

(PHY_CTRL_EXT3.ACTIPHY_SLEEP_TIMER). The actual sleep time duration is random, from –80 ms to +60 ms, to avoid two linked PHYs in ActiPHY mode entering a lock-up state during operation.

After sending signal energy on the relevant media, the PHY returns to the low power state.

#### 4.4.6.2 Link Partner Wake-up State

In the link partner wake-up state, the PHY attempts to wake up the link partner. Up to three complete FLP bursts are sent on alternating pairs A and B of the Cat5 media for a duration based on the wake-up timer, which is set using register bits 20E1.12:11.

After sending signal energy on the relevant media, the PHY returns to the low power state.

#### 4.4.6.3 Normal Operating State

In normal operation, the PHY establishes a link with a link partner. When the media is unplugged or the link partner is powered down, the PHY waits for the duration of the programmable link status time-out timer, which is set using ACTIPHY_LINK_TIMER_MSB_CFG and ACTIPHY_LINK_TIMER_LSB_CFG in the PHY_AUX_CTRL_STAT register. It then enters the low power state.

### 4.4.7 Testing Features

The integrated transceivers include several testing features designed to facilitate performing system-level debugging.

#### 4.4.7.1 Core Voltage and I/O Voltage Monitor

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 device contains a monitoring circuit that provides a readout of the I/O and core supply voltages. The voltage value that is read out is accurate to within ±25 mV for the core and low voltage I/O supplies (0.9 V to 1.4 V) and ±50 mV for the high voltage I/O supplies (2.25 V to 2.75 V).

#### 4.4.7.2 Ethernet Packet Generator (EPG)

The Ethernet Packet Generator (EPG) can be used at each of the 10/100/1000BASE-T speed settings for Copper Cat5 media to isolate problems between the MAC and the PHY, or between a local PHY and its remote link partner. Enabling the EPG feature effectively disables all MAC interface transmit pins and selects the EPG as the source for all data transmitted onto the twisted pair interface.

**Important** The EPG is intended for use with laboratory or in-system testing equipment only. Do not use the EPG testing feature when the PHY is connected to a live network.

To use the EPG feature, set PHY_1000BT_EPG2.EPG_ENA to 1.

When PHY_1000BT_EPG2.EPG_RUN_ENA is set to 1, the PHY begins transmitting Ethernet packets based on the settings in the PHY_1000BT_EPG1 and PHY_1000BT_EPG2 registers. These registers set:

- Source and destination addresses for each packet
- Packet size
- Inter-packet gap
- FCS state
- Transmit duration
- Payload pattern

If PHY_1000BT_EPG1.TRANSMIT_DURATION_CFG is set to 0, PHY_1000BT_EPG1.EPG_RUN_ENA is cleared automatically after 30,000,000 packets are transmitted.

#### 4.4.7.3 CRC Counters

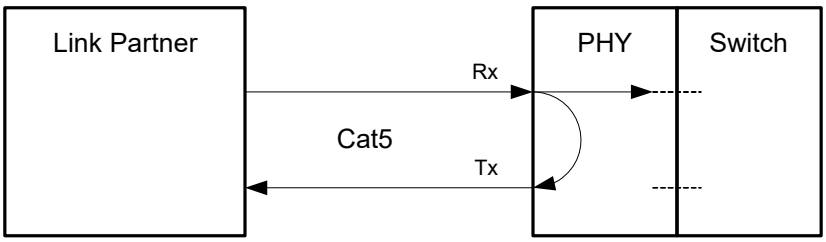
Two separate CRC counters are available in the PHY: a 14-bit good CRC counter available through PHY_CRC_GOOD_CNT.CRC_GOOD_PKT_CNT and a separate 8-bit bad CRC counter in PHY_CTRL_EXT4.CRC_1000BT_CNT.

#### 4.4.7.4 Far-End Loopback

The far-end loopback testing feature is enabled by setting PHY_CTRL_EXT1.FAR_END_LOOPBACK_ENA to 1. When enabled, it forces incoming data from a link

partner on the current media interface, into the MAC interface of the PHY, to be re-transmitted back to the link partner on the media interface as shown in the following illustration. The incoming data also appears on the receive data pins of the MAC interface. Data present on the transmit data pins of the MAC interface is ignored when using this testing feature.

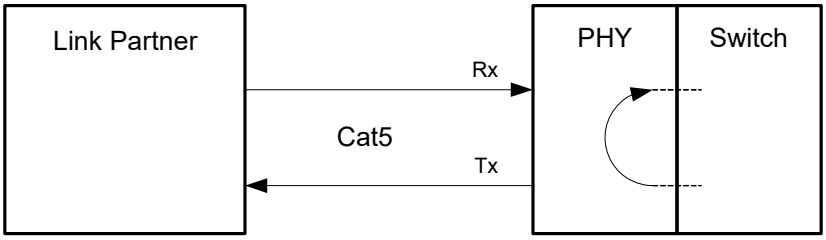
***Figure 13 •*** **Far-End Loopback Diagram**




### 4.4.7.5 Near-End Loopback

When the near-end loopback testing feature is enabled (by setting PHY_CTRL.LOOPBACK_ENA to 1), data on the transmit data pins (TXD) is looped back in the PCS block, onto the device receive data pins (RXD), as shown in the following illustration. When using this testing feature, no data is transmitted over the network.

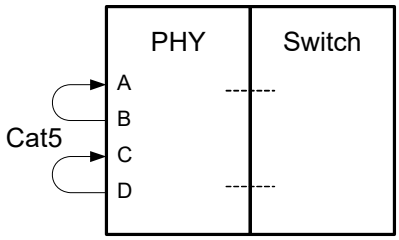
***Figure 14 •*** **Near-End Loopback Diagram**




### 4.4.7.6 Connector Loopback

The connector loopback testing feature allows the twisted pair interface to be looped back externally. When using the connector loopback feature, the PHY must be connected to a loopback connector or a loopback cable. Pair A must be connected to pair B, and pair C to pair D, as shown in the following illustration. The connector loopback feature functions at all available interface speeds.

***Figure 15 •*** **Connector Loopback Diagram**




When using the connector loopback testing feature, the device auto-negotiation, speed, and duplex configuration is set using device registers 0, 4, and 9. For 1000BASE-T connector loopback, the following additional writes are required, executed in the following steps:

1. Enable the 1000BASE-T connector loopback. Set PHY_CTRL_EXT2.CON_LOOPBACK_1000BT_ENA to 1.
2. Disable pair swap correction. Set PHY_CTRL_EXT2.CON_LOOPBACK_1000BT_ENA to 1.

## 4.4.8 VeriPHY™ Cable Diagnostics

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices include a comprehensive suite of cable diagnostic functions that are available through the onboard processor. These functions enable

cable operating conditions and status to be accessed and checked. The VeriPHY suite has the ability to identify the cable length and operating conditions and to isolate common faults that can occur on the Cat5 twisted pair cabling.

For the functional details of the VeriPHY suite and operating instructions, see ENT-AN0125, *PHY, Integrated PHY-Switch VeriPHY - Cable Diagnostics Feature* Application Note.

## 4.5 Statistics

The following table lists the registers for the statistics module.

***Table 19 •*** **Counter Registers**

| Register | Description | Replication |
|---|---|---|
| SYS::STAT:CNT | Data register for reading port counters | Per counter per port |
| SYS::STAT_CFG.STAT_CLEAR_SHOT | Clears port counters | |
| SYS::STAT_CFG.STAT_CLEAR_PORT | Selects which port's counters to clear | |
| SYS::STAT_CFG.TX_GREEN_CNT_MODE<br>SYS::STAT_CFG.TX_YELLOW_CNT_MODE | Controls whether to counts bytes or frames for Tx priority counters | |
| SYS::STAT_CFG.DROP_GREEN_CNT_MODE<br>SYS::STAT_CFG.DROP_YELLOW_CNT_MODE | Controls whether to counts bytes or frames for drop priority counters | |
| ANA::AGENCTRL.GREEN_COUNT_MODE<br>ANA::AGENCTRL.YELLOW_COUNT_MODE<br>ANA::AGENCTRL.RED_COUNT_MODE | Controls whether to counts bytes or frames for Rx priority counters | |

All counters for all ports are sharing a common statistics block with directly addressable counters. Each counter is 32 bits wide, which is large enough to ensure a wrap-around time longer than 13 seconds.

Each switch core port has 43 Rx counters, 18 FIFO drop counters, and 31 Tx counters.

The following table defines the per-port available Rx counters and lists the counter's base address in the common statistics block.

***Table 20 •*** **Rx Counters in the Statistics Block**

| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Rx | c_rx_oct | 0x000 | Received octets in good and bad frames. |
| Rx | c_rx_uc | 0x001 | Number of good unicasts. |
| Rx | c_rx_mc | 0x002 | Number of good multicasts. |
| Rx | c_rx_bc | 0x003 | Number of good broadcasts. |
| Rx | c_rx_short | 0x004 | Number of short frames with valid CRC (<64 bytes). |
| Rx | c_rx_frag | 0x005 | Number of short frames with invalid CRC (<64 bytes). |
| Rx | c_rx_jabber | 0x006 | Number of long frames with invalid CRC (according to MAXLEN.MAX_LENGTH). |
| Rx | c_rx_crc | 0x007 | Number of CRC errors, alignment errors and RX_ER events. |
| Rx | c_rx_sz_64 | 0x008 | Number of 64-byte frames in good and bad frames. |

*Table 20 •* **Rx Counters in the Statistics Block *(continued)***

| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Rx | c_rx_sz_65_127 | 0x009 | Number of 65-127-byte frames in good and bad frames. |
| Rx | c_rx_sz_128_255 | 0x00A | Number of 128-255-byte frames in good and bad frames. |
| Rx | c_rx_sz_256_511 | 0x00B | Number of 256-511-byte frames in good and bad frames. |
| Rx | c_rx_sz_512_1023 | 0x00C | Number of 512-1023-byte frames in good and bad frames. |
| Rx | c_rx_sz_1024_1526 | 0x00D | Number of 1024-1526-byte frames in good and bad frames. |
| Rx | c_rx_sz_jumbo | 0x00E | Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames. |
| Rx | c_rx_pause | 0x00F | Number of received pause frames. |
| Rx | c_rx_control | 0x010 | Number of MAC control frames received. |
| Rx | c_rx_long | 0x011 | Number of long frames with valid CRC (according to MAXLEN.MAX_LENGTH). |
| Rx | c_rx_cat_drop | 0x012 | Number of frames dropped due to classifier rules. |
| Rx | c_rx_red_prio_0 | 0x013 | Number of received frames classified to QoS class 0 and discarded by a policer. |
| Rx | c_rx_red_prio_1 | 0x014 | Number of received frames classified to QoS class 1 and discarded by a policer. |
| Rx | c_rx_red_prio_2 | 0x015 | Number of received frames classified to QoS class 2 and discarded by a policer. |
| Rx | c_rx_red_prio_3 | 0x016 | Number of received frames classified to QoS class 3 and discarded by a policer. |
| Rx | c_rx_red_prio_4 | 0x017 | Number of received frames classified to QoS class 4 and discarded by a policer |
| Rx | c_rx_red_prio_5 | 0x018 | Number of received frames classified to QoS class 5 and discarded by a policer. |
| Rx | c_rx_red_prio_6 | 0x01A | Number of received frames classified to QoS class 6 and discarded by a policer. |
| Rx | c_rx_red_prio_7 | 0x01B | Number of received frames classified to QoS class 7 and discarded by a policer. |
| Rx | c_rx_yellow_prio_0 | 0x01C | Number of received frames classified to QoS class 0 and marked yellowby a policer |
| Rx | c_rx_yellow_prio_1 | 0x01D | Number of received frames classified to QoS class 1 and marked yellow by a policer |
| Rx | c_rx_yellow_prio_2 | 0x01E | Number of received frames classified to QoS class 2 and marked yellow by a policer |
| Rx | c_rx_yellow_prio_3 | 0x01F | Number of received frames classified to QoS class 3 and marked yellow by a policer |
| Rx | c_rx_yellow_prio_4 | 0x020 | Number of received frames classified to QoS class 4 and marked yellow by a policer |

*Table 20 •* **Rx Counters in the Statistics Block *(continued)***

| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Rx | c_rx_yellow_prio_5 | 0x021 | Number of received frames classified to QoS class 5 and marked yellow by a policer |
| Rx | c_rx_yellow_prio_6 | 0x022 | Number of received frames classified to QoS class 6 and marked yellow by a policer |
| Rx | c_rx_yellow_prio_7 | 0x023 | Number of received frames classified to QoS class 7 and marked yellow by a policer |
| Rx | c_rx_green_prio_0 | 0x024 | Number of received frames classified to QoS class 0 and marked green by a policer. |
| Rx | c_rx_green_prio_1 | 0x025 | Number of received frames classified to QoS class 1 and marked green by a policer. |
| Rx | c_rx_green_prio_2 | 0x026 | Number of received frames classified to QoS class 2 and marked green by a policer. |
| Rx | c_rx_green_prio_3 | 0x027 | Number of received frames classified to QoS class 3 and marked green by a policer. |
| Rx | c_rx_green_prio_4 | 0x028 | Number of received frames classified to QoS class 4 and marked green by a policer. |
| Rx | c_rx_green_prio_5 | 0x029 | Number of received frames classified to QoS class 5 and marked green by a policer. |
| Rx | c_rx_green_prio_6 | 0x02A | Number of received frames classified to QoS class 6 and marked green by a policer. |
| Rx | c_rx_green_prio_7 | 0x02B | Number of received frames classified to QoS class 7 and marked green by a policer. |

The following table defines the per-port available FIFO drop counters and lists the counter address.

*Table 21 •* **FIFO Drop Counters in the Statistics Block**

| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Drop | c_dr_local | 0xC00 | Number of frames discarded due to no destinations. |
| Drop | c_dr_tail | 0xC01 | Number of frames discarded due to no more memory in the queue system (tail drop). |
| Drop | c_dr_yellow_prio_0 | 0xC02 | Number of FIFO discarded frames classified to QoS class 0 with DP level 1 |
| Drop | c_dr_yellow_prio_1 | 0xC03 | Number of FIFO discarded frames classified to QoS class 1 with DP level 1 |
| Drop | c_dr_yellow_prio_2 | 0xC04 | Number of FIFO discarded frames classified to QoS class 2 with DP level 1 |
| Drop | c_dr_yellow_prio_3 | 0xC05 | Number of FIFO discarded frames classified to QoS class 3 with DP level 1 |
| Drop | c_dr_yellow_prio_4 | 0xC06 | Number of FIFO discarded frames classified to QoS class 4 with DP level 1 |
| Drop | c_dr_yellow_prio_5 | 0xC07 | Number of FIFO discarded frames classified to QoS class 5 with DP level 1 |

*Table 21 •* **FIFO Drop Counters in the Statistics Block *(continued)***

| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Drop | c_dr_yellow_prio_6 | 0xC08 | Number of FIFO discarded frames classified to QoS class 6 with DP level 1 |
| Drop | c_dr_yellow_prio_7 | 0xC09 | Number of FIFO discarded frames classified to QoS class 7 with DP level 1 |
| Drop | c_dr_green_prio_0 | 0xC0A | Number of FIFO discarded frames classified to QoS class 0 with DP level 0. |
| Drop | c_dr_green_prio_1 | 0xC0B | Number of FIFO discarded frames classified to QoS class 1 with DP level 0. |
| Drop | c_dr_green_prio_2 | 0xC0C | Number of FIFO discarded frames classified to QoS class 2 with DP level 0. |
| Drop | c_dr_green_prio_3 | 0xC0D | Number of FIFO discarded frames classified to QoS class 3 with DP level 0. |
| Drop | c_dr_green_prio_4 | 0xC0E | Number of FIFO discarded frames classified to QoS class 4 with DP level 0. |
| Drop | c_dr_green_prio_5 | 0xC0F | Number of FIFO discarded frames classified to QoS class 5 with DP level 0 |
| Drop | c_dr_green_prio_6 | 0xC10 | Number of FIFO discarded frames classified to QoS class 6 with DP level 0. |
| Drop | c_dr_green_prio_7 | 0xC11 | Number of FIFO discarded frames classified to QoS class 7 with DP level 0. |

The following table defines the per-port available Tx counters and lists the counter address.

*Table 22 •* **Tx Counters in the Statistics Block**

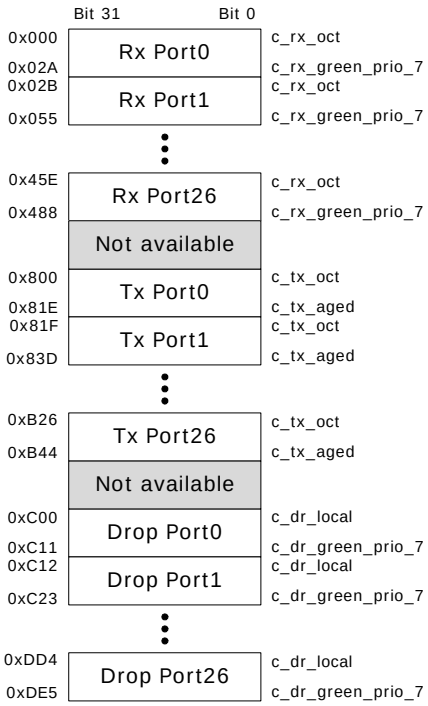
| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Tx | c_tx_oct | 0x800 | Transmitted octets in good and bad frames. |
| Tx | c_tx_uc | 0x801 | Number of good unicasts. |
| Tx | c_tx_mc | 0x802 | Number of good multicasts. |
| Tx | c_tx_bc | 0x803 | Number of good broadcasts. |
| Tx | c_tx_col | 0x804 | Number of transmitted frames experiencing a collision. An excessive collided frame gives 16 counts. |
| Tx | c_txdrop | 0x805 | Number of frames dropped due to excessive collisions or late collisions. |
| Tx | c_txpause | 0x806 | Number of transmitted pause frames in 1 Gbps full-duplex. Transmitted pause frames in 10/100 Mbps full-duplex are not counted. |
| Tx | c_tx_sz_64 | 0x807 | Number of 64-byte frames in good and bad frames. |
| Tx | c_tx_sz_65_127 | 0x808 | Number of 65-127-byte frames in good and bad frames. |
| Tx | c_tx_sz_128_255 | 0x809 | Number of 128-255-byte frames in good and bad frames. |
| Tx | c_tx_sz_256_511 | 0x80A | Number of 256-511-byte frames in good and bad frames. |

*Table 22 •* **Tx Counters in the Statistics Block *(continued)***

| Type | Short Name | Base Address | Description |
|---|---|---|---|
| Tx | c_tx_sz_512_1023 | 0x80B | Number of 512-1023-byte frames in good and bad frames. |
| Tx | c_tx_sz_1024_1526 | 0x80C | Number of 1024-1526-byte frames in good and bad frames. |
| Tx | c_tx_sz_jumbo | 0x80D | Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames. |
| Tx | c_tx_yellow_prio_0 | 0x80E | Number of transmitted frames classified to QoS class 0 with DP level 1. |
| Tx | c_tx_yellow_prio_1 | 0x80F | Number of transmitted frames classified to QoS class 1 with DP level 1. |
| Tx | c_tx_yellow_prio_2 | 0x810 | Number of transmitted frames classified to QoS class 2 with DP level 1. |
| Tx | c_tx_yellow_prio_3 | 0x811 | Number of transmitted frames classified to QoS class 3 with DP level 1. |
| Tx | c_tx_yellow_prio_4 | 0x812 | Number of transmitted frames classified to QoS class 4 with DP level 1. |
| Tx | c_tx_yellow_prio_5 | 0x813 | Number of transmitted frames classified to QoS class 5 with DP level 1. |
| Tx | c_tx_yellow_prio_6 | 0x814 | Number of transmitted frames classified to QoS class 6 with DP level 1. |
| Tx | c_tx_yellow_prio_7 | 0x815 | Number of transmitted frames classified to QoS class 7 with DP level 1. |
| Tx | c_tx_green_prio_0 | 0x816 | Number of transmitted frames classified to QoS class 0 with DP level 0. |
| Tx | c_tx_green_prio_1 | 0x817 | Number of transmitted frames classified to QoS class 1 with DP level 0. |
| Tx | c_tx_green_prio_2 | 0x818 | Number of transmitted frames classified to QoS class 2 with DP level 0. |
| Tx | c_tx_green_prio_3 | 0x819 | Number of transmitted frames classified to QoS class 3 with DP level 0. |
| Tx | c_tx_green_prio_4 | 0x81A | Number of transmitted frames classified to QoS class 4 with DP level 0. |
| Tx | c_tx_green_prio_5 | 0x81B | Number of transmitted frames classified to QoS class 5 with DP level 0. |
| Tx | c_tx_green_prio_6 | 0x81C | Number of transmitted frames classified to QoS class 6 with DP level 0. |
| Tx | c_tx_green_prio_7 | 0x81D | Number of transmitted frames classified to QoS class 7 with DP level 0. |
| Tx | c_tx_aged | 0x81E | Number of frames dropped due to frame aging. |

The counters are placed in a directly addressable RAM as shown in the following illustration.

***Figure 16 •*** **Counter Layout**




The reading of a counter uses direct addressing. The following shows the address to use when reading a given counter for a port:

- Rx counter: Rx counter's base address + 43*port
- Tx counter: Tx counter's base address + 31*port
- Drop counter: Drop counter's base address + 18*port

For information about Rx counter base addresses, see Table 20, page 43. For information about Tx counter base addresses, see Table 22, page 46. For information about drop counter base addresses, see Table 21, page 45.

Writing to register STAT_CFG.STAT_CLEAR_SHOT clears all associated counters in the port module specified in STAT_CFG.STAT_CLEAR_PORT.

It is possible to select whether to count frames or bytes for the following specific counters:

- The Rx priority counters (c_rx_red_prio_*, c_rx_yellow_prio_*, c_rx_green_prio_*, where x is 0 through 7).
- The Tx priority counters (c_tx_yellow_prio_*, c_tx_green_prio_*, where x is 0 through 7).
- The Drop priority counters (c_dr_yellow_prio_*, c_dr_green_prio_*, where x is 0 through 7).

The Rx priority counters are programmed through ANA::AGENCTRL, and the Tx and drop priority counters are programmed through SYS::STAT_CFG. When counting bytes, the frame length excluding inter frame gap and preamble is counted.

For testing purposes, all counters are both readable and writable. All counters wrap around to 0 when reaching the maximum.

For more information about how the counters map to relevant MIBs, see Port Counters, page 191.

## 4.6 Classifier

The switch core includes a common classifier, which determines a number of properties affecting the forwarding of each frame through the switch. These properties are:

- Frame acceptance filtering – Drop illegal frame types.
- QoS classification – Assign one of eight QoS classes to the frame.
- DSCP classification - Assign one of 64 DSCP values to the frame.

- VLAN classification – Extract tag information from the frame or use the port VLAN.
- Link aggregation code generation – Generate the link aggregation code.
- CPU forwarding determination – Determine CPU Forwarding and CPU extraction queue number

The outcome of the classifier is the basic classification result, which can be overruled by more intelligent frame processing in the VCAP-II IS1. For more information, see VCAP-II, page 58.

### 4.6.1 General Data Extraction Setup

This section provides information about the overall settings for data extraction controlling the other tasks in the classifier, VCAP-II, analyzer, and rewriter.

The following table lists the registers associated with general data extraction.

***Table 23 •*** **General Data Extraction Registers**

| Register | Description | Replication |
|---|---|---|
| SYS::PORT_MODE.L3_PARSE_CFG | Enables the use of Layer 3 and 4 protocol information for classification and frame processing. | Per port |
| SYS::VLAN_ETYPE_CFG | Ethernet Type for S-tags in addition to default value 0x88A8. | None |
| ANA:PORT.VLAN_CFG.VLAN_INNER_TAG_ENA | Enables using inner VLAN tag for basic classification if available in incoming frame. | Per port |
| ANA:PORT:S1_VLAN_INNER_TAG_ENA | Enables using inner VLAN tag for IS1 key generation if available in incoming frame. | Per port per IS1 lookup |

In the devices, it is programmable which VLAN tags are recognized. The use of Layer-3 and Layer-4 information for classification and forwarding can also be controlled.

The devices recognize three different VLAN tags:

- Customer tags (C-TAGs), which use TPID 0x8100.
- Service tags (S-TAGs), which use TPID 0x88A8 (IEEE 802.1ad).
- Service tags (S-TAGs), which use a custom TPID programmed in SYS::VLAN_ETYPE_CFG.

The devices can parse and use information from up to two VLAN tags of any of the kinds described above.

By default, the outer VLAN tag is extracted and used for both the basic classification and the VCAP IS1 key generation. However, for both the basic classification and the VCAP IS1, there is an option to use the inner VLAN tag instead for frames with at least two VLAN tags. For basic classification, this is controlled in VLAN_CFG.VLAN_INNER_TAG_ENA and affects both QoS, DP, and VLAN classification as well as the frame acceptance filter. For IS1, this is controlled per lookup in S1_VLAN_INNER_TAG_ENA.

Various blocks in the devices use Layer-3 and Layer-4 information for classification and forwarding. Layer-3 and Layer-4 information can be extracted from a frame with up to two VLAN tags. Frames with more than two VLAN tags are considered non-IP frames.

The actual use of Layer-3 and Layer-4 information for classification, forwarding, and rewriting is enabled in SYS::PORT_MODE.L3_PARSE_CFG. The following blocks are affected by this functionality:

- Basic classification: QoS, DP, and DSCP classification, link aggregation code generation, CPU forwarding
- VCAP-II: TCAM keys (IS1, IS2) using Layer 3 and Layer4 information
- Analyzer: Flooding and forwarding of IP multicast frames
- Rewriter: Rewriting of IP information

## 4.6.2 Frame Acceptance Filtering

The following table lists the registers associated with frame acceptance filtering.

***Table 24 •*** **Frame Acceptance Filtering Registers**

| Register | Description | Replication |
|---|---|---|
| PORT::PORT_MISC | Configures forwarding of special frames | Per port |
| ANA:PORT:DROP_CFG | Configures discarding of illegal frame types | Per port |

Based on the configurations in the DROP_CFG and PORT_MISC registers, the classifier instructs the queue system to drop or forward certain frames types, such as:

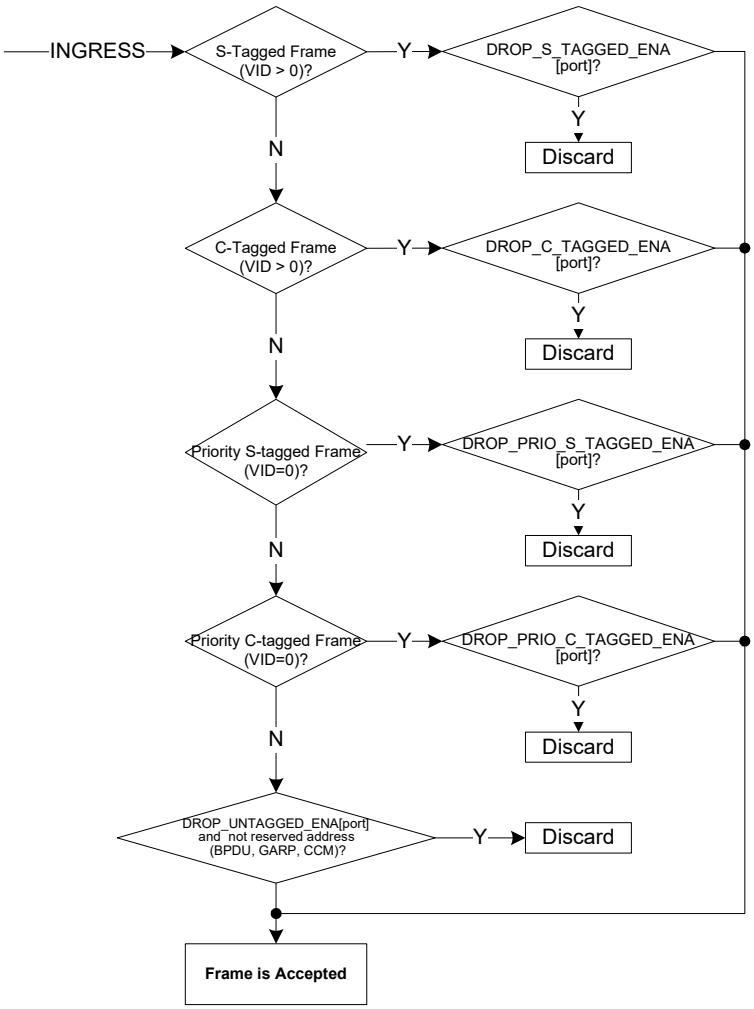
- Frames with a multicast source MAC address
- Frames with a null source or null destination MAC address (address = 0x000000000000)
- Frames with errors signaled by the MAC (for example, an FCS error)
- MAC control frames
- Pause frames after flow control processing in the MAC.
- Untagged frames (excluding frames with reserved destination MAC addresses from the BPDU, GARP, and Link trace/CCM address ranges).
- Priority S-tagged frames
- Priority C-tagged frames
- VLAN S-tagged frames
- VLAN C-tagged frames

By default, MAC control frames, pause frames, and frames with errors are dropped by the classifier.

The VLAN acceptance filter decides whether a frame's VLAN tagging is allowed on the port. By default, the outer VLAN tag is used as input to the filter, however, there is an option to use the inner VLAN tag instead for double tagged frames (VLAN_CFG.VLAN_INNER_TAG_ENA).

The following illustration shows the flowchart for the VLAN acceptance filter.

*Figure 17 •* **VLAN Acceptance Filter**




If the frame is accepted by the VLAN acceptance filter, it can still be discarded in other places of the switch, such as:

- Policers, due to traffic exceeding a peak information rate.
- IS2 Security TCAM, due to permit/deny rules.
- Analyzer, due to forwarding decisions such as VLAN ingress filtering.
- Queue system, due to lack of resources, frame aging, or excessive collisions.

## 4.6.3 QoS, DP, and DSCP Classification

This section provides information about the functions in the QoS, DP, and DSCP classification. The three tasks are described one, because the tasks have a significant amount of functionality in common.

The following table lists the registers associated with QoS, DP, and DSCP classification.

*Table 25 •* **QoS, DP, and DSCP Classification Registers**

| Register | Description | Replication |
|---|---|---|
| ANA.PORT.QOS_CFG | Configuration of the overall classification flow for QoS, DP, and DSCP. | Per port |
| ANA:PORT:QOS_PCP_DEI_MAP_CFG | Mapping from (DEI, PCP) to (DP, QoS). | Per port per DEI per PCP |

*Table 25 •* **QoS, DP, and DSCP Classification Registers *(continued)***

| Register | Description | Replication |
|---|---|---|
| ANA::DSCP_CFG | DSCP configuration per DSCP value. | Per DSCP |
| ANA::DSCP_REWR_CFG | DSCP rewrite values per DP level and QoS class. | Per DP and per QoS |

The basic classification provides the user with control of the QoS, DP, and DSCP classification algorithm. The result of the basic classification are the following frame properties, which follow the frame through the switch:

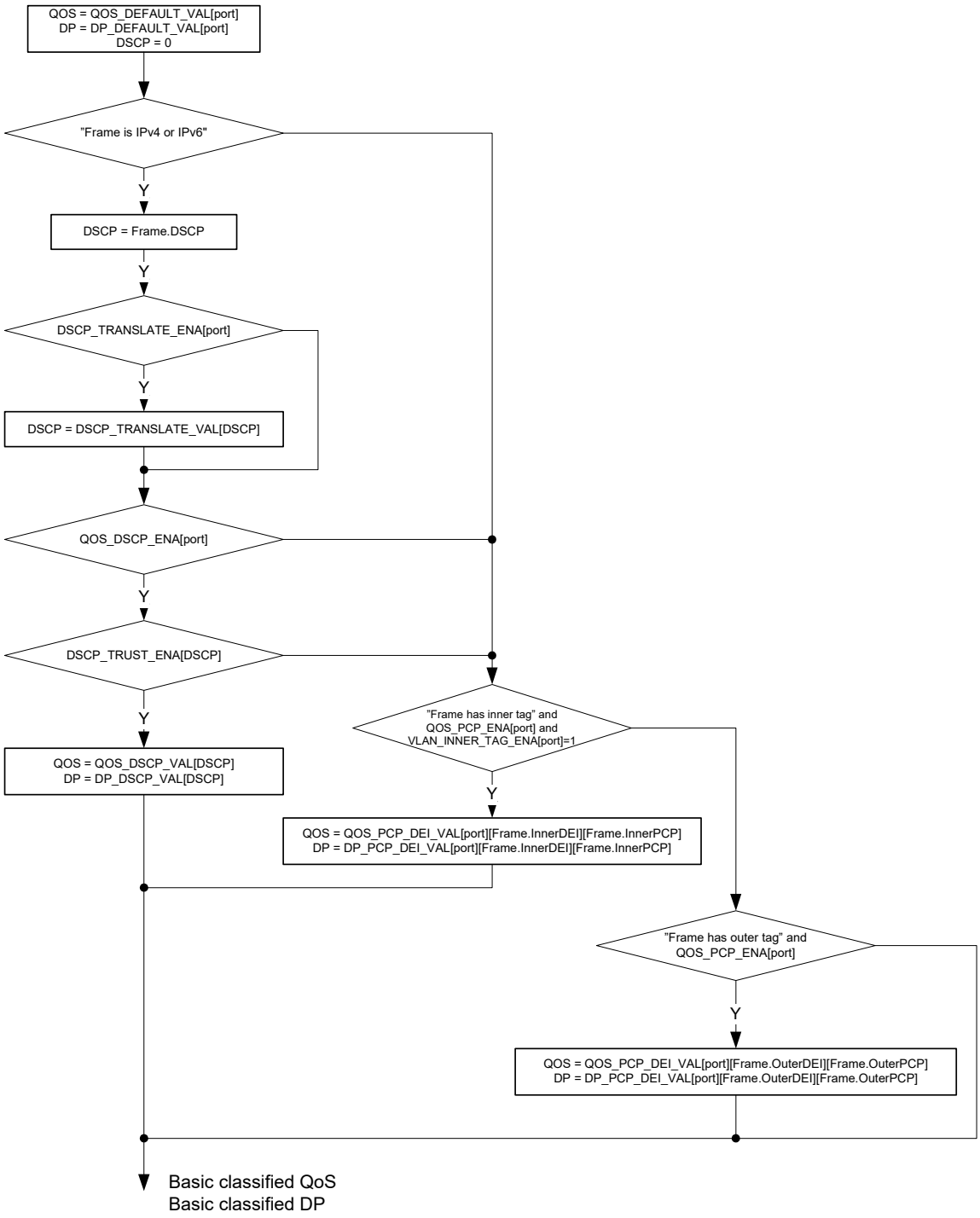
- The frame's QoS class. This class is encoded in a 3-bit field, where 7 is the highest priority QoS class and 0 is the lowest priority QoS class. The QoS class is used by the queue system when enqueuing frames and when evaluating resource consumptions, for policing, statistics, and rewriter actions.
- The frame's DP level. This level is encoded in a 1-bit field, where frames with DP = 1 have the highest probability of being dropped and frames with DP = 0 have the lowest probability. The DP level is used by the MEF compliant policers for measuring committed and peak information rates, for restricting memory consumptions in the queue system, for collecting statistics, and for rewriting priority information in the rewriter. The DP level is incremented by the policers if a frame is exceeding a programmed committed information rate.
- The frame's DSCP. This value is encoded in a 6-bit fields. The DSCP value is forwarded with the frame to the rewriter where it is translated and rewritten into the frame. The DSCP value is only applicable to IPv4 and IPv6 frames.

The classifier looks for the following fields in the incoming frame to determine the QoS, DP, and DSCP classification:

- Port default QoS class and DP level. The default DSCP value is the frame's DSCP value. For non-IP frames, the DSCP is 0 and it not used elsewhere in the switch.
- Priority Code Point (PCP) when the frame is VLAN tagged or priority tagged. There is an option to use the inner tag for double tagged frames (VLAN_CFG.VLAN_INNER_TAG_ENA). Both S-tagged and C-tagged frames are considered.
- Drop Eligible Indicator (DEI) when the frame is VLAN tagged or priority tagged. There is an option to use the inner tag for double tagged frames (VLAN_CFG.VLAN_INNER_TAG_ENA). Both S-tagged and C-tagged frames are considered.
- DSCP (all 6 bits, both for IPv4 and IPv6 packets). The classifier can look for the DSCP value behind up to two VLAN tags.

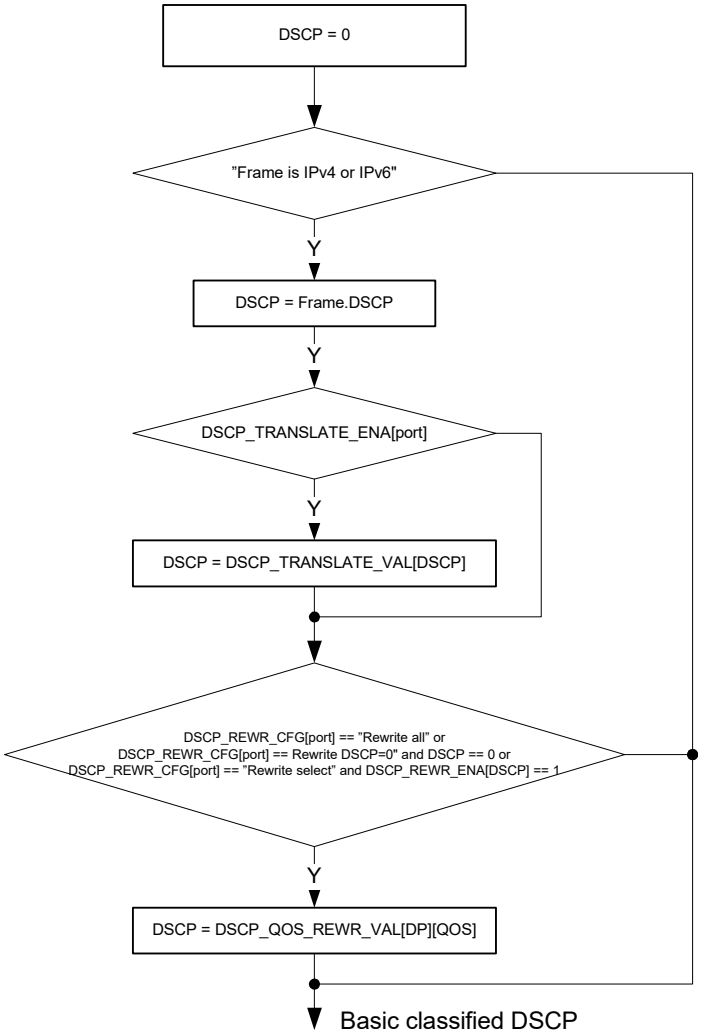
The following illustration shows the flow chart of basic QoS and DP classification.

***Figure 18 •*** **QoS and DP Basic Classification Flow Chart**




The following illustration shows the flow chart for basic DSCP classification.

***Figure 19 •*** **Basic DSCP Classification Flow Chart**




The translation part of the DSCP classification is common for both QoS, DP and DSCP classification.

The basic classified QoS, DP, and DSCP can be overwritten by more intelligent decisions made in the VCAP IS1.

### 4.6.4 VLAN Classification

The following table lists the registers associated with VLAN classification.

***Table 26 •*** **VLAN Configuration Registers**

| Register | Description | Replication |
|---|---|---|
| ANA:PORT:VLAN_CFG | Configures the port's processing of VLAN information in VLAN-tagged and priority-tagged frames. Configures the port-based VLAN. | Per port |

The VLAN classification determines a tag header for all frames. The tag header includes the following information:

- Priority Code Point (PCP)
- Drop Eligible Indicator (DEI)
- VLAN Identifier (VID)
- Tag Protocol Identifier (TPID) type (TAG_TYPE). This field informs whether tag used for classification was a C-tag or an S-tag.

The tag header determined by the classifier is carried with the frame through the switch and is used in various places such as the analyzer for forwarding and the rewriter for egress tagging operations.

The devices recognize three kinds of tags based on the TPID, which is the EtherType in front of the tag:

- Customer tags (C-TAGs), which use TPID 0x8100.
- Service tags (S-TAGs), which use TPID 0x88A8 (IEEE 802.1ad).
- Service tags (S-TAGs), which use a custom TPID programmed in SYS::VLAN_ETYPE_CFG.

For customer tags and service tags, both VLAN tags (tags with nonzero VID) and priority tags (tags with VID = 0) are processed.

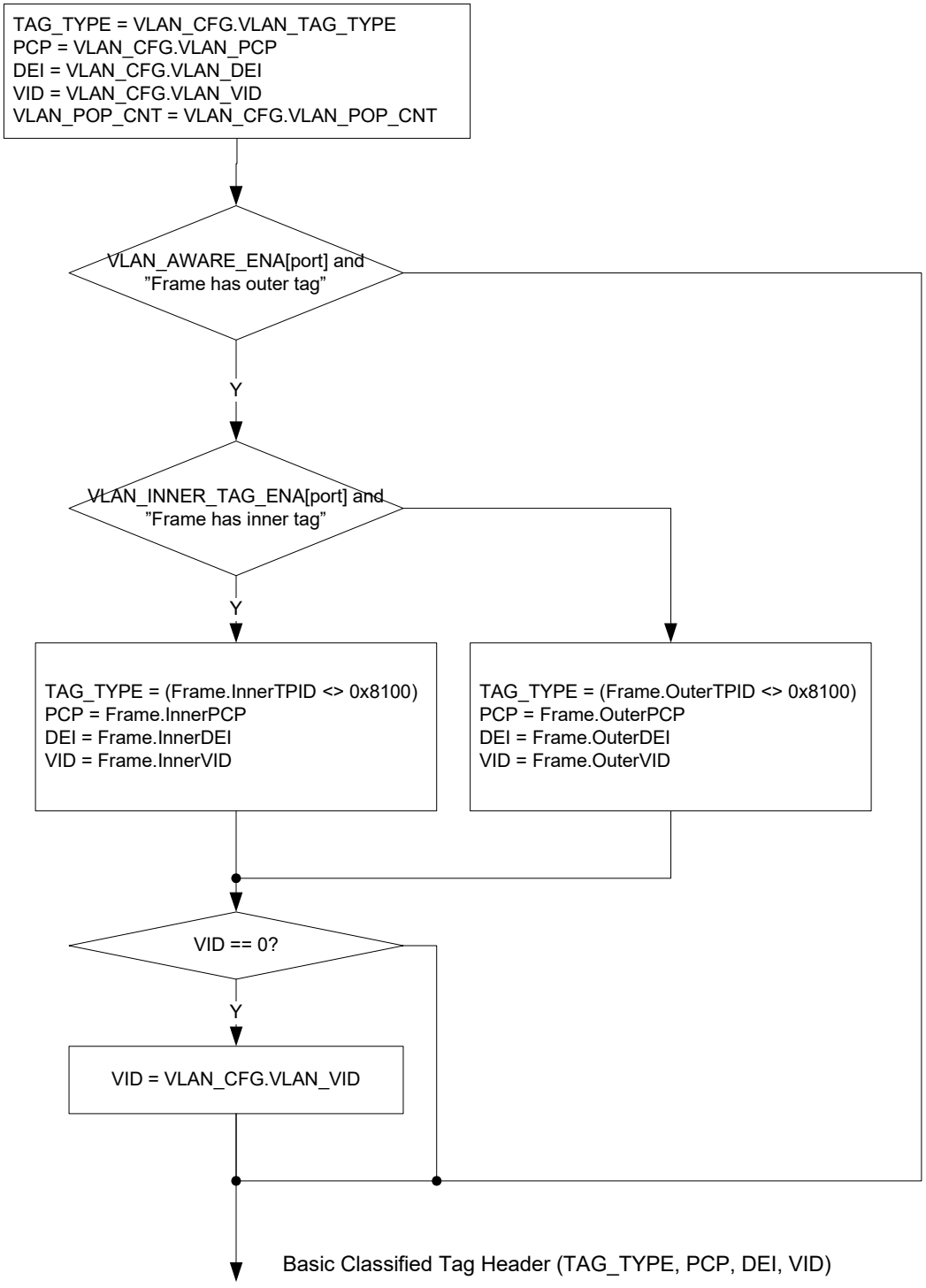
The tag header is either retrieved from a tag in the incoming frame or from a default port-based tag header. The port-based tag header is configured in ANA:PORT:VLAN_CFG.

For double tagged frames, there is an option to use the inner tag instead of the outer tag (VLAN_CFG.VLAN_INNNER_TAG_ENA).

In addition to the tag header, the ingress port decides the number of VLAN tags to pop at egress (VLAN_POP_CNT). If the configured number of tags to pop is greater than the actual number of tags in the frame, the number is reduced to the number of actual tags in the frame.

The following illustration shows the flow chart for basic VLAN classification.

***Figure 20 •*** **Basic VLAN Classification Flow Chart**




The basic classified tag header can be overwritten by more intelligent decisions made in the VCAP IS1.

## 4.6.5 Link Aggregation Code Generation

This section provides information about the functions in link aggregation code generation.

The following table lists the registers associated with aggregation code generation.

*Table 27 •* **Aggregation Code Generation Registers**

| Register | Description | Replication |
|---|---|---|
| ANA::AGGR_CFG | Configures use of Layer-2 through Layer-4 flow information for link aggregation code generation. | Common |

The classifier generates a link aggregation code, which is used in the analyzer when selecting to which port in a link aggregation group a frame is forwarded.

The following contributions to the link aggregation code is configured in the AGGR_CFG register:

- Destination MAC address—use the lower 12 bits of the DMAC.
- Source MAC address—use the lower 12 bits of the SMAC.
- IPv6 flow label—use the 20 bits of the flow label.
- IPv4 source and destination IP addresses—use the lower 8 bits of the SIP and DIP.
- TCP/UDP source and destination port for IPv4 and IPv6 frames—use the lower 8 bits of the SPORT and DPORT.
- Random aggregation code—use a pseudo-random number instead of the frame information.

Each of the enabled contributions are XOR'ed together, yielding a 4-bit aggregation code ranging from 0 to 15. For more information about how the aggregation code is used, see Link Aggregation, page 217.

## 4.6.6 CPU Forwarding Determination

The following table lists the registers associated with CPU forwarding.

*Table 28 •* **CPU Forwarding Determination**

| Register | Description | Replication |
|---|---|---|
| CPU_FWD_CFG | Enables CPU forwarding for various frame types | Per port |
| CPU_FWD_BPDU_CFG | Enables CPU forwarding per BPDU address | Per port |
| CPU_FWD_GARP_CFG | Enables CPU forwarding per GARP address | Per port |
| CPU_FWD_CCM_CFG | Enables CPU forwarding per CCM/Link trace address | Per port |
| CPUQ_CFG | CPU extraction queues for various frame types | None |
| CPUQ_8021_CFG | CPU extraction queues for BPDU, GARP, and CCM addresses. | None |

The classifier has support for determining whether certain frames must be forwarded to the CPU extraction queues. Other parts of the device can also determine CPU forwarding, for example, the analyzer, based on MAC table entries or the VCAP IS2. All events leading to CPU forwarding are OR'ed together, and the final CPU extraction queue mask, which is available to the user, contains the sum of all events leading to CPU extraction. For more information, see CPU Extraction and Injection, page 235.

Upon CPU forwarding by the classifier, the frame type determines whether the frame is redirected or copied to the CPU. Any frame type or event causing a redirection to the CPU cause all front ports to be removed from the forwarding decision - only the CPU receives the frame. When copying a frame to the CPU, the normal forwarding of the frame is unaffected.

The following table lists the frame types, with respect to CPU forwarding, that are recognized by the classifier.

***Table 29 •*** **Frame Type Definitions for CPU Forwarding**

| Frame | Condition | Copy/Redirect |
|---|---|---|
| BPDU frames. Reserved Addresses (IEEE 802.1D 7.12.6) | DMAC = 0x0180C2000000 to 0x0180C20000F (BPDUs and various Slow protocols supporting spanning tree, link aggregation, port authentication) | Redirect |
| Reserved ALLBRIDGE address | DMAC = 0x0180C2000010 | Redirect |
| GARP Application Addresses (IEEE 802.1D 12.5) | DMAC = 0x0180C2000020 to 0x0180C200002F | Redirect |
| CCM/Link Trace Addresses (IEEE P802.1ag) | DMAC = 0x0180C2000030 to 0x0180C200003F | Redirect |
| IGMP | DMAC = 0x01005E000000 to 0x01005E7FFFFF<br>EtherType = IPv4<br>IP Protocol = IGMP | Redirect |
| MLD | DMAC = 0x333300000000 to 0x3333FFFFFFFF<br>EtherType = IPv6<br>IPv6 Next Header = 0<br>Hop-by-hop options header with the first option being a Router Alert option with the MLD message (Option Type = 5, Opt Data Len = 2, Option Data = 0). | Redirect |
| IPv4 Multicast Ctrl | DMAC = 0x01005E000000 to 0x01005E7FFFFF<br>EtherType = IPv4<br>IP protocol is not IGMP<br>IPv4 DIP inside 224.0.0.x | Copy |
| Source port | All frames received on enabled ingress port | Copy |
| All other frames | | |

## 4.7 VCAP-II

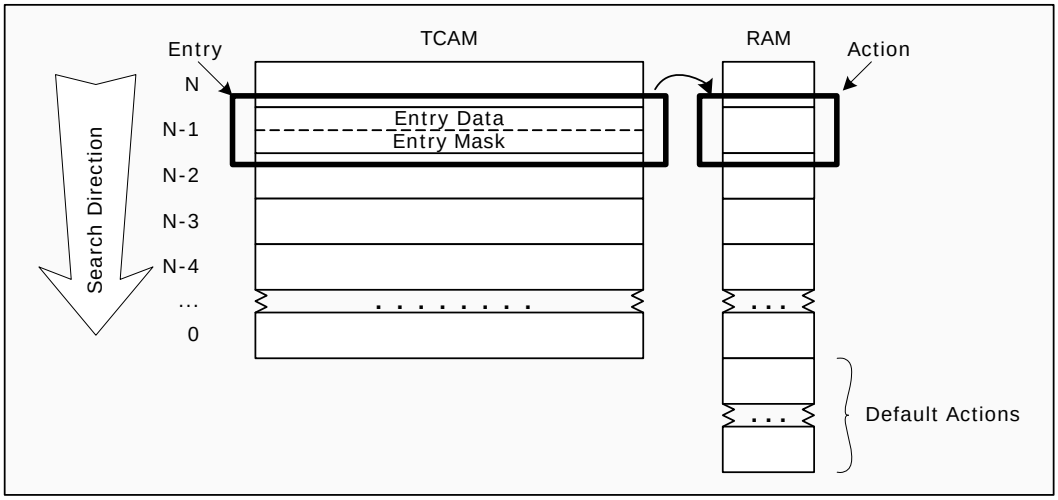
The VCAP-II is a second generation content-aware packet processor for wire-speed packet inspection for rich implementation of, for example, advanced VLAN and QoS classifications and manipulations, IP source guarding, and security features for wireline and wireless applications.

The following describes the three VCAPs implemented in the devices: IS1, IS2, and ES0. IS1 and IS2 are ingress VCAPs working on the incoming frames while ES0 is an egress VCAP working on all outgoing frames.

When a VCAP is enabled, each frame is examined to determine the frame type (for example IPv4 TCP frame) so that the frame information is extracted according to the frame type. Together with port-specific configuration and classification results from the basic classification, the extracted frame information makes up an entry key, which is passed to a TCAM and matched against entries in the TCAM.

An entry in the TCAM consists of a pattern and a mask, where the mask allows pattern-matching with the use of “don’t cares”. The first matching entry is then used to select an action. The following illustration provides a functional overview of a general TCAM.

***Figure 21 •*** **VCAP Functional Overview**




Each frame results in five ingress VCAP lookups and one egress lookup per destination port. The lookups use different keys and the results determine the frame's ingress classification, security handling, and egress VLAN manipulation. The five ingress lookups and the associated VCAPs are:

1. Advanced ingress classification, first lookup
   VCAP: IS1
   Key: IS1
   Entry: IS1 Control Entry
2. Advanced ingress classification, second lookup
   VCAP: IS1
   Key: IS1
   Entry: IS1 Control Entry
3. IP source guarding check
   VCAP: IS1
   Key: SMAC_SIP4 (IPv4 frames) or SMAC_SIP6 (IPv6 frames)
   Entry: SMAC_SIP4 Control Entry or SMAC_SIP6 Control Entry
4. Security enforcement, first lookup
   VCAP: IS2
   Key: MAC_ETYPE, MAC_LLC, MAC_SNAP, ARP, IP4_OTHER, IP4_TCP_UDP, or IP6_STD, depending on frame type
   Entry: Access Control Entry
5. Security enforcement, second lookup
   VCAP: IS2
   Key: MAC_ETYPE, MAC_LLC, MAC_SNAP, ARP, IP4_OTHER, IP4_TCP_UDP, or IP6_STD, depending on frame type
   Entry: Access Control Entry

The egress lookup per destination port and associated VCAP is:

1. Egress tagging and frame manipulations
   VCAP: ES0
   Key: ES0
   Entry: Egress Control Entry

The IP source guarding check is only carried out for IP frames.

CPU injected frames are subject to all the above VCAP lookups in IS1 and IS2, and the ES0 lookup is not performed.

Each frame is classified to one of six overall VCAP frame types. The frame type determines the information to extract from the frame and also which VCAP entries to match against. The following table lists which frame types are used and which VCAP entries the frame types are matched against in IS1 and

IS2. Note that a lookup in ES0 is independent of the frame type and all frames match against all entries in the TCAM.

***Table 30 •*** **VCAP Frame Types**

| Frame Type | Condition | IS1 Entries | IS2 Entries |
|---|---|---|---|
| IPv6 Frame | The Type/Len field is equal to 0x86DD.<br>The IP version is 6.<br>Special IPv6 frames:<br>•IPv6 TCP frame: Next header is TCP (0x6)<br>•IPv6 UDP frame: Next header is UDP (0x11)<br>•IPv6 Other frame: Next header is neither TCP nor UDP | Frame type flags:<br>ETYPE_LEN = 1<br>IP_SNAP = 1<br>IP4 = 0<br>TCP_UDP<br>TCP | IP6_STD |
| IPv4 Frame | The Type/Len field is equal to 0x800.<br>The IP version is 4.<br>Special IPv4 frames:<br>•IPv4 TCP frame: IP protocol is TCP (0x6)<br>•IPv4 UDP frame: IP protocol is UDP (0x11)<br>•IPv4 Other frame: IP protocol is neither TCP nor UDP | Frame type flags:<br>ETYPE_LEN = 1<br>IP_SNAP = 1<br>IP4 = 1<br>TCP_UDP<br>TCP | IP4_TCP_UDP<br>IP4_OTHER |
| (R)ARP Frame | The Type/Len field is equal to 0x0806 (ARP) or 0x8035 (RARP). | Frame type flags:<br>ETYPE_LEN = 1<br>IP_SNAP = 0 | ARP |
| SNAP Frame | The Type/Len field is less than 0x600.<br>The Destination Service Access Point field, DSAP is equal to 0xAA.<br>The Source Service Access Point field, SSAP is equal to 0xAA.<br>The Control field is equal to 0x3. | Frame type flags:<br>ETYPE_LEN = 0<br>IP_SNAP = 1 | MAC_SNAP |
| LLC Frame | The Type/Len field is less than 0x600<br>The LLC header does not indicate a SNAP frame. | Frame type flags:<br>ETYPE_LEN = 0<br>IP_SNAP = 0 | MAC_LLC |
| ETYPE Frame | The Type/Len field is greater than or equal to 0x600.<br>The Type field does not indicate any of the previously mentioned frame types, that is, ARP, RARP, IPv4, or IPv6. | Frame type flags:<br>ETYPE_LEN = 1<br>IP_SNAP = 0 | MAC_ETYPE |

In addition, Precision Time Protocol (PTP) frames are handled specifically by IS2. The following encapsulations of PTP frames are supported:

- PTP over Ethernet:
  ETYPE frame with Type/Len = 0x88F7.
  Matched against MAC_ETYPE entries.
- PTP over UDP over IPv4:
  IPv4 UDP frame with UDP destination port numbers 319 or 320.
  Matched against IP4_TCP_UDP entries.
- PTP over UDP over IPv6
  IPv6 UDP frame with UDP destination port numbers 319 or 320.
  Matched against IP6_STD entries or IP4_TCP_UDP when IP6_STD entries are disabled. For more information, see Port Configuration, page 61.

For PTP over Ethernet, the following PTP fields are always extracted:

- TransportSpecific (byte 0)
- MessageType (byte 0)
- VersionPTP (byte 1)

In addition, bytes 2-7 following the EtherType can be extracted when source MAC address overloading is used. For more information, see Port Configuration, page 61.

**Note** Byte 0 is the byte immediately following the EtherType, then byte 1, byte 2, and so on.

For PTP over UDP, the following PTP fields are always extracted:

- messageType (byte 0)
- domainNumber (byte 4)
- flagField: flags 1, 2, and 7 (byte 6)

In addition, the bytes 0, 1, 4, and 6 following the UDP header can be extracted when source IP address overloading is used.

**Note** Byte 0 is the byte immediately following the EtherType, then byte 1, byte 2, and so on.

## 4.7.1 Port Configuration

This section provides information about special port configurations that control the key generation for the VCAPs.

The following table lists the registers associated with port configuration for VCAP.

***Table 31 •*** **Port Module Configuration of VCAP**

| Register | Description | Replication |
|---|---|---|
| ANA:PORT:VCAP_CFG | Configuration of the key generation for the VCAPs | Per port |
| REW:PORT:PORT_CFG | Enables VCAP ES0 | Per port |

Each port module affects the key generation for VCAPs IS1 and IS2 through the VCAP_CFG registers, and the rewriter affects VCAP ES0 through the REW:PORT:PORT_CFG.ES0_ENA register.

### 4.7.1.1 VCAP IS1 Port Configuration

The following port configurations are available for IS1:

- Enable lookups in IS1 (VCAP_CFG.S1_ENA). If disabled, frames received by the port module are not matched against rules in VCAP IS1.
- Use destination information rather than source information (VCAP_CFG.S1_DMAC_DIP_ENA). By default, the two advanced classification lookups in IS1 use the source MAC address and source IP address from the incoming frame when generating the key. Through S1_DMAC_DIP_ENA, the corresponding destination information, destination MAC address, and destination IP address can be used instead. This can be controlled per lookup so that, for example, the first lookup applies source information, and the second applies destination information.
- Use inner VLAN tag rather than outer VLAN tag (VCAP_CFG.S1_VLAN_INNER_TAG_ENA). By default, the two advanced classification lookups in IS1 use the outer VLAN tag from the incoming frame when generating the key. Through S1_VLAN_INNER_TAG_ENA, the inner tag for double tagged frames can be used. This can be controlled per lookup so that, for example, the first lookup applies the outer tag, and the second lookup applies the inner tag. For single tagged frames, the outer VLAN tag is always used.

### 4.7.1.2 VCAP IS2 Port Configuration

The following port configurations are available for IS2:

- Enable lookups in IS2 (VCAP_CFG.S2_ENA). If disabled, frames received by the port module are not matched against rules in VCAP IS2.
- Default PAG value (VCAP_CFG.PAG_VAL). This PAG value is the initial value. Actions out of IS1 can change the PAG value before it is used in the key for IS2.
- Source IP address overloading (VCAP_CFG.S2_UDP_PAYLOAD_ENA). If enabled, UDP payload overwrites the source IP address for IP4_TCP_UDP entry types in IS2. The UDP payload is bytes 0, 1, 4, and 6 following the UDP header. This is controllable per lookup.

- Source MAC address overloading (VCAP_CFG.S2_ETYPE_PAYLOAD_ENA). If enabled, frame payload bytes overwrites the source MAC address for MAC_ETYPE entry types in IS2. The frame payload used is bytes 2 through 7 following the EtherType. This is controllable per lookup.

Each port module can control a hierarchy of which entry types in IS2 to use for different frame types. For instance, it is controllable whether IPv6 frames are matched against IP6_STD entries, IP4_TCPUDP entries, or MAC_ETYPE entries. Note that matching against an entry type also controls how the key is generated.

With reference to the VCAP_CFG register, the following table lists the hierarchy for different frame types.

***Table 32 •*** **Hierarchy of IS2 Entry Types**

| Frame Type | Description |
|---|---|
| IPv6 Frames | Configuration: S2_IP6_STD_DIS and S2_IP6_TCPUDP_OTHER_DIS.<br>If S2_IP6_STD_DIS is cleared, IPv6 frames are matched against IP6_STD entries.<br>If S2_IP6_STD_DIS is set and S2_IP6_TCPUDP_OTHER_DIS is cleared, IPv6 frames are matched against IP4_TCPUDP or IP4_OTHER entries.<br>If both are set, IPv6 frames are matched against MAC_ETYPE entries. |
| IPv4 TCP and UDP frames | Configuration: S2_IP_TCPUDP_DIS<br>If S2_IP_TCPUDP_DIS is cleared, IPv4 TCP and UDP frames are matched against IP4_TCPUDP entries.<br>If S2_IP_TCPUDP_DIS is set, IPv4 TCP and UDP frames are matched against MAC_ETYPE entries. |
| IPv4 Other frames (non-TCP and non-UDP) | Configuration: S2_IP_OTHER_DIS<br>If S2_IP_OTHER_DIS is cleared, IPv4 Other frames are matched against IP4_OTHER entries.<br>If S2_IP_OTHER_DIS is set, IPv4 Other frames are matched against MAC_ETYPE entries. |
| ARP frames | Configuration: S2_ARP_DIS<br>If S2_ARP_DIS is cleared, ARP frames are matched against MAC_ETYPE entries.<br>If S2_ARP_DIS is set, ARP frames are matched against MAC_ETYPE entries. |
| SNAP frames | Configuration: S2_SNAP_DIS<br>If S2_SNAP_DIS is cleared, SNAP frames are matched against LLC entries.<br>If S2_SNAP_DIS is set, SNAP frames are matched against LCC entries. |

#### 4.7.1.3 Port Configuration of VCAP ES0

The rewriter configures VCAP ES0 through REW:PORT:PORT_CFG.ES0_ENA. If ES0 is disabled, frames transmitted on the port are not matched against rules in ES0.

## 4.7.2 VCAP IS1

This section provides information about the IS1 key, the SMAC_SIP4 key, the SMAC_SIP6 key, and associated actions.

#### 4.7.2.1 IS1 Entry Key Encoding

All frame types are subject to the two IS1 lookups. The same key is used for all frame types, however, within the key there are frame type flags that indicate the originating frame type. In addition, certain key

fields are overloaded with different frame fields depending on the frame type flag settings. The following table lists the IS1 key.

***Table 33 •*** **IS1 Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Match Information** | | | |
| IS1_TYPE | 0 | 1 | Cleared for IS1 lookups and set for SMAC_SIP6 lookups. |
| FIRST | 1 | 1 | Set for first lookup and cleared for second lookup. |
| **Interface Information** | | | |
| IGR_PORT_MASK | 2 | 27 | Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port. |
| **Tagging Information** | | | |
| VLAN_TAGGED | 29 | 1 | Set if frame has one or more Q-tags. Independent of port VLAN awareness. |
| VLAN_DBL_TAGGED | 30 | 1 | Set if frame has two or more Q-tags. Independent of port VLAN awareness. |
| TPID | 31 | 1 | 0: Customer TPID<br>1: Service TPID (88A8 or programmable)<br>TPID is derived from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA. |
| VID | 32 | 12 | Frame's VID if frame is tagged, otherwise port default.<br>VID is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA. |
| DEI | 44 | 1 | Frame's DEI if frame is tagged, otherwise port default.<br>DEI is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA. |
| PCP | 45 | 3 | Frame's PCP if frame is tagged, otherwise port default.<br>PCP is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA. |
| **Layer-2 Information** | | | |
| L2_SMAC_HIGH | 48 | 16 | Frame's source MAC address, bits 47:32.<br>Use destination MAC address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port. |
| L2_SMAC_LOW | 64 | 32 | Frame's source MAC address, bits 31:0.<br>Use destination MAC address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port. |
| L2_MC | 96 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1) |
| L2_BC | 97 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF) |
| IP_MC | 98 | 1 | Set if frame is IPv4 frame and frame's destination MAC address is an IPv4 multicast address (0x01005E0 /25).<br>Set if frame is IPv6 frame and frame's destination MAC address is an IPv6 multicast address (0x3333 /16). |
| ETYPE_LEN | 99 | 1 | Frame type flag.<br>Set if frame has EtherType >= 0x600 (Frame is type encoded). Otherwise cleared (Frame is length encoded). |

*Table 33 •* **IS1 Key *(continued)***

| Field name | Bit | Width | Description |
|---|---|---|---|
| ETYPE | 100 | 16 | Overloaded field for different frame types:<br>LLC frame: ETYPE = [DSAP, SSAP]<br>SNAP frame: ETYPE = PID[4:3]<br>IPv4 or IPv6 TCP/UDP frame: ETYPE = DPORT<br>IPv4 or IPv6 Other frame: ETYPE = IP protocol<br>ARP or ETYPE frame: ETYPE = Frame's EtherType. |
| IP_SNAP | 116 | 1 | Frame type flag.<br>Set if frame is IPv4, IPv6, or SNAP frame. |
| IP4 | 117 | 1 | Frame type flag.<br>Set if frame is IPv4 frame |
| **Layer-3 Information** | | | |
| L3_FRAGMENT | 118 | 1 | Set if IPv4 frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0).<br>Layer 4 information cannot not be trusted. |
| L3_OPTIONS | 119 | 1 | Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed.<br>Layer 4 information cannot not be trusted. |
| L3_DSCP | 120 | 6 | Frame's DSCP value. The DSCP value may have been translated during basic classification, see QoS, DP, and DSCP Classification, page 51. |
| L3_IP4_SIP | 126 | 32 | Overloaded fields for different frame types:<br>LLC frame: L3_IP4_SIP = [CTRL, PAYLOAD[0:2]]<br>SNAP frame: L3_IP4_SIP = [PID[2:0], PAYLOAD[0]]<br>IPv4 or IPv6 frame: L3_IP4_SIP = source IP address, bits [31:0]<br>ARP or ETYPE frame: L3_IP4_SIP = PAYLOAD[0:3]<br>For IPv4 or IPv6 frames, use destination IP address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port. |
| **Layer-4 Information** | | | |
| TCP_UDP | 158 | 1 | Frame type flag.<br>Set if frame is IPv4/IPv6 TCP or UDP frame. |
| TCP | 159 | 1 | Frame type flag.<br>Set if frame is IPv4/IPv6 TCP frame. |
| L4_SPORT | 160 | 16 | TCP/UDP frame's source port. |
| L4_RNG | 176 | 8 | Range mask with one bit per range. A bit is set, if the corresponding range is matched.<br>Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP<br>Input to range checkers:<br>– SPORT/DPORT: From frame<br>– VID: From frame if tagged, otherwise port's VID<br>– DSCP: Translated DSCP from the basic classification.<br>See section Range Checkers, page 77. |

Fields not applicable to a certain frame type (for example, L3_OPTIONS for an IPv6 frame) must be set to don't care for entries the frame type can match.

If L3_FRAGMENT or L3_OPTIONS are set to 1 or set to don't care, Layer 4 information cannot be trusted and should be set to don't-care for such entries.

#### 4.7.2.2 SMAC_SIP6 Entry Key Encoding

All IPv6 frames are subject to a SMAC_SIP6 lookup. The following table lists the SMAC_SIP6 key.

***Table 34 •*** **SMAC_SIP6 Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Lookup Information** | | | |
| IS1_TYPE | 0 | 1 | Cleared for IS1 lookups and set for SMAC_SIP6 lookups. |
| **Interface Information** | | | |
| IGR_PORT | 1 | 5 | The port number where the frame was received (0-26). |
| **Layer-2 Information** | | | |
| L2_SMAC_HIGH | 6 | 16 | Frame's source MAC address, bits 47:32. |
| L2_SMAC_LOW | 22 | 32 | Frame's source MAC address, bits 31:0. |
| **Layer-3 Information** | | | |
| L3_IP6_SIP_3 | 54 | 32 | Frame's source IPv6 address, bits 127:96. |
| L3_IP6_SIP_2 | 86 | 32 | Frame's source IPv6 address, bits 95:64. |
| L3_IP6_SIP_1 | 118 | 32 | Frame's source IPv6 address, bits 63:32. |
| L3_IP6_SIP_0 | 150 | 32 | Frame's source IPv6 address, bits 31:0. |

### 4.7.2.3 SMAC_SIP4 Entry Key Encoding

All IPv4 frames are subject to a SMAC_SIP4 lookup. The following table lists the SMAC_SIP4 key.

***Table 35 •*** **SMAC_SIP4 Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Interface Information** | | | |
| IGR_PORT | 0 | 5 | The port number where the frame was received (0-26). |
| **Layer-2 Information** | | | |
| L2_SMAC_HIGH | 5 | 16 | Frame's source MAC address, bits 47:32. |
| L2_SMAC_LOW | 21 | 32 | Frame's source MAC address, bits 31:0. |
| **Layer-3 Information** | | | |
| L3_IP4_SIP | 53 | 32 | Frame's source IPv4 address. |

### 4.7.2.4 IS1, SMAC_SIP4, and SMAC_SIP6 Action Encoding

The VCAP generates an action vector from each of the two IS1 lookups and, for IP frames, from the SMAC_SIP6 or SMAC_SIP4 lookups. The action vectors are combined into one action vector, which is applied to the classification of the frame.

There are no default action vectors for the IS1.

The following table lists the available fields for the IS1 action vector.

***Table 36 •*** **IS1 Action Fields**

| Action field | Bit | Width | Description |
|---|---|---|---|
| DSCP_ENA | 0 | 1 | If set, use DSCP_VAL as classified DSCP value. Otherwise, DSCP value from basic classification is used. |
| DSCP_VAL | 1 | 6 | See DSCP_ENA. |

*Table 36 •* **IS1 Action Fields *(continued)***

| Action field | Bit | Width | Description |
|---|---|---|---|
| DP_ENA | 7 | 1 | If set, use DP_VAL as classified drop precedence level. Otherwise, drop precedence level from basic classification is used. |
| DP_VAL | 8 | 1 | See DP_ENA. |
| QOS_ENA | 9 | 1 | If set, use QOS_VAL as classified QoS class. Otherwise, QoS class from basic classification is used. |
| QOS_VAL | 10 | 3 | See QOS_ENA. |
| PAG_ENA | 13 | 1 | If set, use PAG_VAL as policy association group (PAG) input to IS2. Otherwise, PAG from ANA:PORT:VCAP_CFG.PAG_VAL is used. |
| PAG_VAL | 14 | 8 | See PAG_ENA. |
| VID_REPLACE_ENA | 22 | 1 | Controls the classified VID: VID_REPLACE_ENA=0: Add VID_ADD_VAL to classified VID and use result as new classified VID.<br>VID_REPLACE_ENA = 1: Replace classified VID with VID_VAL value and use as new classified VID. |
| VID_ADD_VAL | 23 | 12 | See VID_REPLACE_ENA. |
| FID_SEL | 35 | 2 | Controls the Filter Identifier (FID) used when looking up the MAC table.<br>0: Disabled: FID = classified VID.<br>1: Use FID_VAL for SMAC lookup in MAC table.<br>2: Use FID_VAL for DMAC lookup in MAC table.<br>3: Use FID_VAL for DMAC and SMAC lookup in MAC table. |
| FID_VAL | 37 | 12 | See FID_SEL. |
| PCP_DEI_ENA | 49 | 1 | If set, use PCP_VAL and DEI_VAL as classified PCP and DEI values. Otherwise, PCP and DEI from basic classification are used. |
| PCP_VAL | 50 | 3 | See PCP_DEI_ENA. |
| DEI_VAL | 53 | 1 | See PCP_DEI_ENA. |
| VLAN_POP_CNT_ENA | 54 | 1 | If set, use VLAN_POP_CNT as the number of VLAN tags to pop from the incoming frame. This number is used by the Rewriter. Otherwise, VLAN_POP_CNT from ANA:PORT:VLAN_CFG.VLAN_POP_CNT is used. |
| VLAN_POP_CNT | 55 | 2 | See VLAN_POP_CNT_ENA. |
| HOST_MATCH | 57 | 1 | Used for IP source guarding. If set, it signals that the host is a valid (for instance a valid combination of source MAC address and source IP address). HOST_MATCH is input to the IS2 key. |
| HIT_STICKY | | 1 | If set, a frame has matched against the associated entry. |

The following table lists the available fields for the SMAC_SIP4 and SMAC_SIP6 actions.

***Table 37 •*** **IS1 SMAC_SIP4 and SMAC_SIP6 Action Fields**

| Action field | Bit | Width | Description |
|---|---|---|---|
| HOST_MATCH | 0 | 1 | Used for IP source guarding. If set, it signals that the host is a valid (for instance a valid combination of source MAC address and source IP address). HOST_MATCH is input to the IS2 key. |
| HIT_STICKY | | 1 | If set, a frame has matched against the associated entry. |

The two IS1 action vectors are applied in two steps. First, the action vector from the first lookup is applied, then the action vector from the second lookup is applied. This implies that if both the first and the second lookup return an action of DP_ENA = 1, for example, the DP_VAL from the second lookup is used. With respect to VID_REPLACE_ENA and VID_VAL, both first and second lookup can add to the classified VID if both action vectors have VID_REPLACE_ENA cleared and VID_VAL > 0.

The action HOST_MATCH is returned by both action vectors from IS1 and by the SMAC_SIP4 and SMAC_SIP6 action vectors. The resulting HOST_MATCH is the inputs OR'ed together so that a host is valid if at least one action vectors has HOST_MATCH = 1.

## 4.7.3 VCAP IS2

This section provides information about the IS2 keys and associated actions.

### 4.7.3.1 IS2 Entry Key Encoding

All frame types are subject to the two IS2 lookups. The frame type determines the key entry type. For more information about VCAP frame types, see Table 30, page 60. The following illustration shows which entry fields are available for each frame type (indicated by the field IS2_TYPE), and the following tables list how the key that is matched against these fields are generated for each of the frame types.

***Figure 22 •*** **IS2 Entry Type Overview**

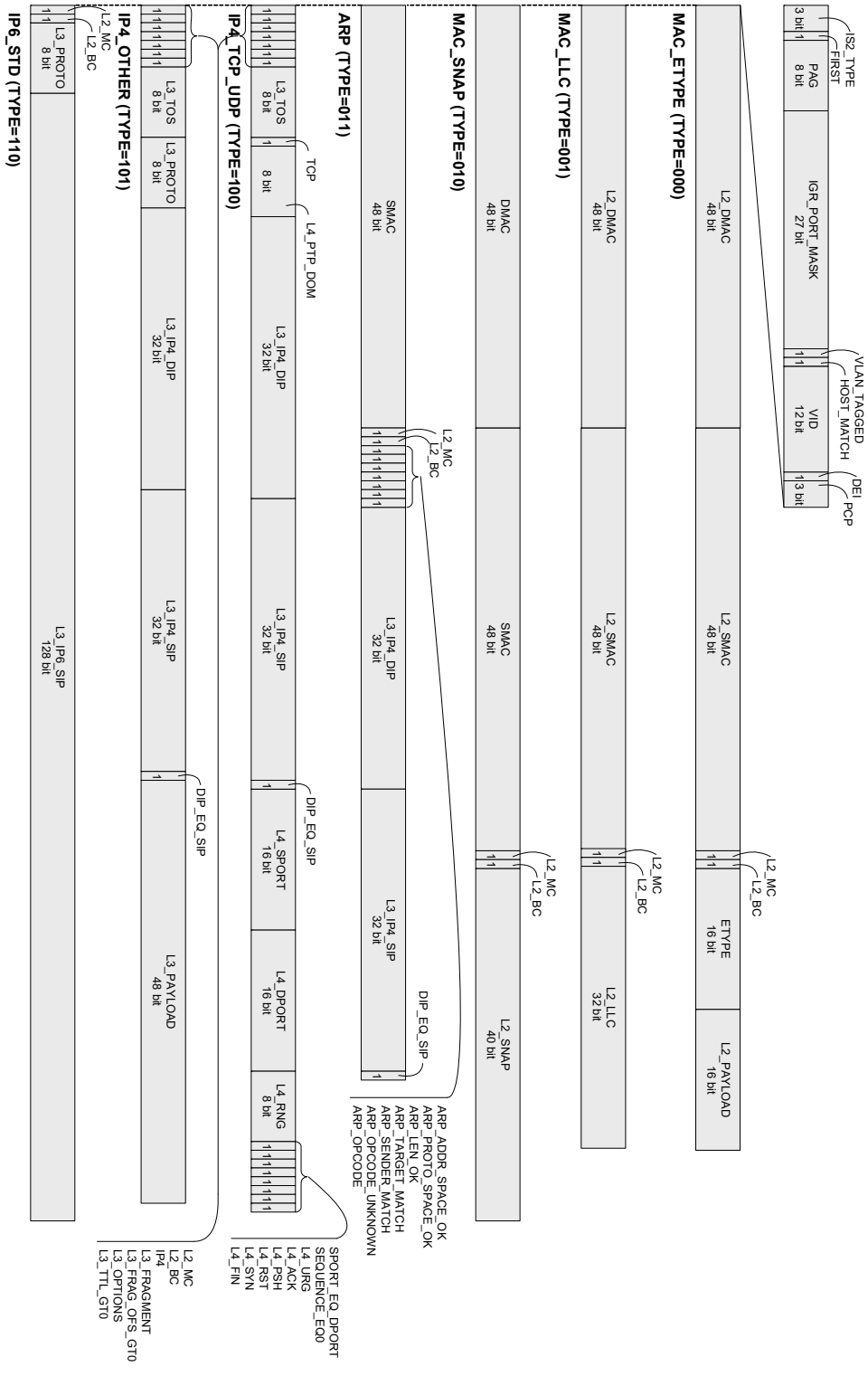


*Table 38 •* **IS2 Common Key Fields**

| Field name | Bit | Width | Description |
|---|---|---|---|
| Lookup Information | | | |
| IS2_TYPE | 0 | 3 | 0: MAC ETYPE entries<br>1: MAC LLC entries<br>2: MAC SNAP entries<br>3: ARP entries<br>4: IPv4 TCP/UDP entries<br>5: IPv4 OTHER entries<br>6: IPv6 STD entries |
| FIRST | 3 | 1 | Set for first lookup and cleared for second lookup. |
| **Interface Information** | | | |
| PAG | 4 | 8 | Policy association group. Action from VCAP IS1. |
| IGR_PORT_MASK | 12 | 27 | Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port. |
| **Tagging and IP Source Guarding Information** | | | |
| VLAN_TAGGED | 39 | 1 | Set if frame has one or more Q-tags. Independent of port VLAN awareness. |
| HOST_MATCH | 40 | 1 | The combined action from the IS1, SMAC_SIP4, and SMAC_SIP6 lookups. Used for IP source guarding. |
| VID | 41 | 12 | Classified VID which is the result of the VLAN classification in basic classification and IS1. |
| DEI | 53 | 1 | Classified DEI which is the final result of the VLAN classification in basic classification and IS1. |
| PCP | 54 | 3 | Classified PCP which is the final result of the VLAN classification in basic classification and IS1. |

*Table 39 •* **IS2 MAC_ETYPE Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_DMAC_HIGH | 57 | 16 | Frame's destination MAC address, bits 47:32. |
| L2_DMAC_LOW | 73 | 32 | Frame's destination MAC address, bits 31:0. |
| L2_SMAC_HIGH | 105 | 16 | Frame's source MAC address, bits 47:32.<br>If ANA:PORT:VCAP_CFG.S2_ETYPE_PAYLOAD_ENA[lookup] is enabled, use payload bytes 2-3 after the frame's EtherType instead of SMAC. |
| L2_SMAC_LOW | 121 | 32 | Frame's source MAC address, bits 31:0.<br>If ANA:PORT:VCAP_CFG.S2_ETYPE_PAYLOAD_ENA[lookup] is enabled, use payload bytes 4-7 after the frame's EtherType instead of SMAC. |
| L2_MC | 153 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |

*Table 39 •* **IS2 MAC_ETYPE Key *(continued)***

| Field name | Bit | Width | Description |
|---|---|---|---|
| L2_BC | 154 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| ETYPE | 155 | 16 | Frame's EtherType. This is the EtherType after up to two VLAN tags. |
| L2_PAYLOAD | 171 | 16 | Payload bytes 0-1 after the frame's EtherType. |

*Table 40 •* **IS2 MAC_LLC Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_DMAC_HIGH | 57 | 16 | Frame's destination MAC address, bits 47:32. |
| L2_DMAC_LOW | 73 | 32 | Frame's destination MAC address, bits 31:0. |
| L2_SMAC_HIGH | 105 | 16 | Frame's source MAC address, bits 47:32. |
| L2_SMAC_LOW | 121 | 32 | Frame's source MAC address, bits 31:0. |
| L2_MC | 153 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 154 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| L2_LLC | 155 | 32 | LLC header and data after up to two VLAN tags and the type/length field. |

*Table 41 •* **IS2 MAC_SNAP Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_DMAC_HIGH | 57 | 16 | Frame's destination MAC address, bits 47:32. |
| L2_DMAC_LOW | 73 | 32 | Frame's destination MAC address, bits 31:0. |
| L2_SMAC_HIGH | 105 | 16 | Frame's source MAC address, bits 47:32. |
| L2_SMAC_LOW | 121 | 32 | Frame's source MAC address, bits 31:0. |
| L2_MC | 153 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 154 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| L2_SNAP | 155 | 40 | SNAP header after LLC header (AA-AA-03). |

*Table 42 •* **IS2 ARP Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_SMAC_HIGH | 57 | 16 | Frame's source MAC address, bits 47:32. |

*Table 42 •* **IS2 ARP Key *(continued)***

| Field name | Bit | Width | Description |
|---|---|---|---|
| L2_SMAC_LOW | 73 | 32 | Frame's source MAC address, bits 31:0. |
| L2_MC | 105 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 106 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| **Layer-3 Information** | | | |
| ARP_ADDR_SPACE_OK | 107 | 1 | Set if hardware address is Ethernet. |
| ARP_PROTO_SPACE_OK | 108 | 1 | Set if protocol address space is IP. |
| ARP_LEN_OK | 109 | 1 | Set if hardware address length = 6 (Ethernet) and IP address length = 4 (IP). |
| ARP_TARGET_MATCH | 110 | 1 | Target hardware address = SMAC (RARP). |
| ARP_SENDER_MATCH | 111 | 1 | Sender hardware address = SMAC (ARP). |
| ARP_OPCODE_UNKNOWN | 112 | 1 | Set if ARP opcode is none of the below are mentioned. |
| ARP_OPCODE | 113 | 2 | 0: ARP request<br>1: ARP reply.<br>2: RARP request.<br>3: RARP reply. |
| L3_IP4_DIP | 115 | 32 | Target IPv4 address. |
| L3_IP4_SIP | 147 | 32 | Sender IPv4 address. |
| DIP_EQ_SIP | 179 | 1 | Set if sender IP address is equal to target IP address. |

*Table 43 •* **IS2 IP4_TCP_UDP Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_MC | 57 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 58 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| **Layer-3 and Layer-4 Information** | | | |
| IP4 | 59 | 1 | Set if frame is IPv4 frame.<br>IPv6 frames can also use IP4_TCP_UDP entries when IP6_STD entries are disabled. |
| L3_FRAGMENT | 60 | 1 | Set if IP frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0). |
| L3_FRAG_OFS_GT0 | 61 | 1 | Set if IP frame is fragmented and it is not the first fragment (Fragments Offset > 0). Such frames do not carry Layer-4 information all Layer-4 information fields in the key are automatically set to don't-care when generating the key. |

*Table 43 •* **IS2 IP4_TCP_UDP Key *(continued)***

| Field name | Bit | Width | Description |
|---|---|---|---|
| L3_OPTIONS | 62 | 1 | Set if IP frame contains options (IP len > 5). IP options are not skipped nor parsed which implies that Layer-4 information cannot be used. All Layer-4 information fields in the key are automatically set to don't-care when generating the key. |
| L3_TTL_GT0 | 63 | 1 | Set if IP TTL is greater than 0. |
| L3_TOS | 64 | 8 | IP TOS field. The DSCP part is the final result from basic classification and IS1. |
| TCP | 72 | 1 | Set if IP Proto = 6 (TCP). |
| L4_PTP_DOM | 73 | 8 | PTP over UDP: domainNumber. |
| L3_IP4_DIP | 81 | 32 | IPv4 frames: Destination IPv4 address.<br>IPv6 frames: Source IPv6 address, bits 63:32. |
| L3_IP4_SIP | 113 | 32 | If UDP frame and VCAP_CFG.S2_UDP_PAYLOAD_ENA[lookup] = 1: Bytes 0, 1, 4, and 6 after the UDP header.<br>Otherwise for IPv4 frames: Source IPv4 address.<br>Otherwise for IPv6 frames: Source IPv6 address, bit 31:0. |
| DIP_EQ_SIP | 145 | 1 | Set if source IP address is equal to destination IP address. |
| L4_DPORT | 146 | 16 | TCP/UDP destination port. |
| L4_SPORT | 162 | 16 | TCP/UDP source port. |
| L4_RNG | 178 | 8 | Range mask with one bit per range. A bit is set, if the corresponding range is matched.<br>Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP.<br>Input to range checkers:<br>– SPORT, DPORT: From frame<br>– VID, DSCP: Classified result from IS1<br>See Range Checkers, page 77. |
| SPORT_EQ_DPORT | 186 | 1 | Set if UDP or TCP source port equals UDP or TCP destination port. |
| SEQUENCE_EQ0 | 187 | 1 | TCP: Set if TCP sequence number is 0.<br>PTP over UDP: messageType bit 0. |
| L4_FIN | 188 | 1 | TCP: TCP flag FIN.<br>PTP over UDP: messageType bit 1. |
| L4_SYN | 189 | 1 | TCP: TCP flag SYN.<br>PTP over UDP: messageType bit 2. |
| L4_RST | 190 | 1 | TCP: TCP flag RST.<br>PTP over UDP: messageType bit 3. |
| L4_PSH | 191 | 1 | TCP: TCP flag PSH.<br>PTP over UDP: flagField bit 1 (twoStepFlag). |
| L4_ACK | 192 | 1 | TCP: TCP flag ACK.<br>PTP over UDP: flagField bit 2 (unicastFlag). |
| L4_URG | 193 | 1 | TCP: TCP flag URG.<br>PTP over UDP: flagField bit 7 (reserved). |

Frames with IP options (L3_OPTIONS set to 1 in key) or fragmented frames, which are not the initial fragment (L3_FRAG_OFS_GT0 set to 1 in key), do not carry Layer-4 information. The Layer-4 fields in

the key (L4_SPORT, L4_DPORT, L4_RNG, SPORT_EQ_DPORT, SEQUENCE_EQ0, L4_FIN, L4_SYN, L4_RST, L4_PSH, L4_ACK, and L4_URG) are automatically set to don't care.

***Table 44 •*** **IS2 IP4_OTHER Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_MC | 57 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 58 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| **Layer-3 Information** | | | |
| IP4 | 59 | 1 | Set if frame is IPv4 frame.<br>IPv6 frames can also use IP4_OTHER entries when IP6_STD entries are disabled. |
| L3_FRAGMENT | 60 | 1 | Set if IP frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0) |
| L3_FRAG_OFS_GT0 | 61 | 1 | Set if IP frame is fragmented and if it is not the first fragment (Fragments Offset > 0). |
| L3_OPTIONS | 62 | 1 | Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed, which implies that L3_PAYLOAD contains data from the IP options for IPv4 frames with IP options. |
| L3_TTL_GT0 | 63 | 1 | Set if IP TTL is greater than 0. |
| L3_TOS | 64 | 8 | IP TOS field. The DSCP part is the final result from basic classification and IS1. |
| L3_PROTO | 72 | 8 | IPv4: IP protocol.<br>IPv6: next header. |
| L3_IP4_DIP | 80 | 32 | IPv4 frames: Destination IPv4 address.<br>IPv6 frames: Source IPv6 address, bits 63:32. |
| L3_IP4_SIP | 112 | 32 | IPv4 frames: Source IPv4 address.<br>IPv6 frames: Source IPv6 address, bit 31:0. |
| DIP_EQ_SIP | 144 | 1 | Set if source IP address is equal to destination IP address. |
| L3_PAYLOAD | 145 | 48 | Bytes 0-5 after IP header. |

***Table 45 •*** **IS2 IP6_STD Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Layer-2 Information** | | | |
| L2_MC | 57 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 58 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |
| **Layer-3 Information** | | | |
| L3_PROTO | 59 | 8 | IPv6 next header. |
| L3_IP6_SIP_3 | 67 | 32 | Frame's source IPv6 address, bits 127:96. |

*Table 45 •* **IS2 IP6_STD Key *(continued)***

| Field name | Bit | Width | Description |
|---|---|---|---|
| L3_IP6_SIP_2 | 99 | 32 | Frame's source IPv6 address, bits 95:64. |
| L3_IP6_SIP_1 | 131 | 32 | Frame's source IPv6 address, bits 63:32. |
| L3_IP6_SIP_0 | 163 | 32 | Frame's source IPv6 address, bits 31:0. |

#### 4.7.3.2 IS2 Action Encoding

The VCAP generates an action vector from each of the two IS2 lookups for each frame.

The first IS2 lookup returns a default action vector per ingress port when no entries are matched, and the second IS2 lookup returns a common default action vector when no entries are matched. There are no difference between an action vector from a match and a default action vector.

The following table lists the available fields for the action vector.

*Table 46 •* **IS2 Action Fields**

| Action field | Bit | Width | Description |
|---|---|---|---|
| HIT_ME_ONCE | 0 | 1 | Setting this bit to 1 causes the first frame that hits this action where the HIT_CNT counter is zero to be copied to the CPU extraction queue specified in CPU_QU_NUM. The HIT_CNT counter is then incremented and any frames that hit this action later are not copied to the CPU. To re-enable the HIT_ME_ONCE functionality, the HIT_CNT counter must be cleared. |
| CPU_COPY_ENA | 1 | 1 | Setting this bit to 1 causes all frames that hit this action to be copied to the CPU extraction queue specified in CPU_QU_NUM. |
| CPU_QU_NUM | 2 | 3 | Determines the CPU extraction queue that is used when a frame is copied to the CPU due to a HIT_ME_ONCE or CPU_COPY_ENA action. |
| MASK_MODE | 5 | 2 | Controls how PORT_MASK is applied.<br>0: No action from PORT_MASK<br>1: Permit/deny (PORT_MASK AND'ed with destination set)<br>2: Policy forwarding (DMAC lookup replaced with PORT_MASK)<br>3: Redirect (SRC, AGGR, VLAN, DMAC lookup replaced with PORT_MASK).<br>The CPU port is never touched by MASK_MODE. |
| MIRROR_ENA | 7 | 1 | Setting this bit to 1 causes frames to be mirrored to the mirror target port (ANA::MIRRPORPORTS) |
| LRN_DIS | 8 | 1 | Setting this bit to 1 disables learning of frames hitting this action. |
| POLICE_ENA | 9 | 1 | Setting this bit to 1 causes frames that hit this action to be policed by the ACL policer specified in POLICE_IDX. Only applies to the first lookup. |
| POLICE_IDX | 10 | 8 | Selects policer index used when policing frames (POLICE_ENA). |
| PORT_MASK | 18 | 26 | Port mask applied to the forwarding decision based on MASK_MODE. |

*Table 46 •* **IS2 Action Fields *(continued)***

| Action field | Bit | Width | Description |
|---|---|---|---|
| PTP_ENA | 44 | 2 | PTP_ENA[0] (One-step): If set, the correction field in PTP header is updated with the residence time.<br>PTP_ENA[1] (Two-step): If set, the egress timestamp information is enqueued in the timestamp queue. |
| HIT_CNT | | 32 | A statistics counter that is incremented by one each time the given action is hit. |

The two action vectors from the first and second lookups are combined into one action vector, which is applied in the analyzer. For more information, see Forwarding Engine, page 92. The actions are combined as follows:

- HIT_ME_ONCE, CPU_COPY_ENA, CPU_QU_NUM:
  If any of the two action vectors have HIT_ME_ONCE or CPU_COPY_ENA set, CPU_COPY_ENA is forwarded to the analyzer. The settings in the action vector from second lookup takes precedence with respect to the CPU extraction queue number.
- MIRROR_ENA:
  If any of the two action vectors have MIRROR_ENA set, MIRROR_ENA is forwarded to the analyzer.
- LRN_DIS:
  If any of the two action vectors have LRN_DIS set, LRN_DIS is forwarded to the analyzer.
- PTP_ENA:
  The settings in the action vector from the second lookup takes precedence if PTP_ENA[0] or PTP_ENA[1] are set.
- POLICE_ENA, POLICE_IDX:
  Only applies to actions from the first lookup.

The following table lists the combinations for MASK_MODE and PORT_MASK when combining actions from the first and second lookups.

*Table 47 •* **MASK_MODE and PORT_MASK Combinations**

| | **Second Lookup** | | | |
|---|---|---|---|---|
| **First Lookup** | **No action** | **Permit/deny** | **Policy** | **Redirect** |
| **No action** | No action | Permit<br>P[(1)] = P2[(2)] | Policy<br>P = P2 | Redirect<br>P = P2 |
| **Permit/deny** | Permit<br>P = P1[(3)] | Permit<br>P = P1 and P2 | Policy<br>P = P1 and P2 | Redirect<br>P = P2 |
| **Policy** | Policy<br>P = P1 | Policy<br>P = P1 and P2 | Policy<br>P = P1 and P2 | Redirect<br>P = P2 |
| **Redirect** | Redirect<br>P = P1 | Redirect<br>P = P1 and P2 | Redirect<br>P = P1 and P2 | Redirect<br>P = P2 |

1. P: Resulting PORT_MASK to analyzer.
2. P2: PORT_MASK from second match.
3. P1: PORT_MASK from first match.

Policy forwarding for frames matching an IPv4 and IPv6 multicast entry in the MAC table is not possible. Policy forwarding is handled as a permit/deny action for such frames.

## 4.7.4 VCAP ES0

This section provides information about the ES0 key and associated actions.

#### 4.7.4.1 ES0 Entry Key Encoding

All frames are subject to one ES0 lookup per destination port, except for frames injected by the CPU port module, which are not matched against ES0 entries. The key in ES0 is independent of frame types. The following table lists the ES0 key.

***Table 48 •*** **ES0 VID Key**

| Field name | Bit | Width | Description |
|---|---|---|---|
| **Interface Information** | | | |
| EGR_PORT | 0 | 5 | The port number where the frame is transmitted (0-26). |
| IGR_PORT | 5 | 5 | The port number where the frame was received (0-26). |
| **Tagging Information** | | | |
| VID | 10 | 12 | Classified VID that is the result of the VLAN classification in basic classification and IS1. |
| DEI | 22 | 1 | Classified DEI that is the final result of the VLAN classification in basic classification and IS1. |
| PCP | 23 | 3 | Classified PCP that is the final result of the VLAN classification in basic classification and IS1. |
| **Layer-2 Information** | | | |
| L2_MC | 26 | 1 | Set if frame's destination MAC address is a multicast address (bit 40 = 1). |
| L2_BC | 27 | 1 | Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF). |

#### 4.7.4.2 ES0 Action Encoding

The VCAP generates one action vector from the ES0 lookup. The lookup returns a default action vector per egress port when no entries are matched. There are no difference between an action vector from a match and a default action vector.

The following table lists the available action fields for ES0. For more information about how the actions are applied to the VLAN manipulations, see VLAN Editing, page 116.

***Table 49 •*** **ES0 Action Fields**

| Action field | Bit | Width | Description |
|---|---|---|---|
| VLD | 0 | 1 | Valid bit, set if entry is in use. |
| TAG_ES0 | 1 | 2 | Control ES0 tagging.<br>0: No ES0 tagging.<br>1: Push ES0 tag only, overrules port settings.<br>2: Push port tag as outer tag if enabled for port and push ES0 as inner tag.<br>3: Always push port tag as outer tag and ES0 as inner tag. |
| TAG_TPID_SEL | 3 | 2 | Selects TPID for ES0 tag.<br>0: 0x8100.<br>1: 0x88A8.<br>2: custom PORT_TPID.<br>3: If IFH.TAG.TAG_TYPE = 0 then 0x8100 else custom.<br>When "No ES0 Tagging" is set for TAG_ES0:<br>0: Push Port tag if enabled for the egress port.<br>1: No port tagging.<br>2-3: Reserved. |

*Table 49 •* **ES0 Action Fields *(continued)***

| Action field | Bit | Width | Description |
|---|---|---|---|
| TAG_VID_SEL | 5 | 2 | Selects VID source for ES0 tag.<br>0: IFH.TAG.VID + VID_B_VAL.<br>1: VID_A_VAL.<br>2: VID_B_VAL.<br>3: REW:PORT:PORT_VLAN_CFG.PORT_VID. |
| VID_A_VAL | 7 | 12 | See TAG_VID_SEL. |
| VID_B_VAL | 19 | 12 | See TAG_VID_SEL. |
| QOS_SRC_SEL | 31 | 2 | Selects the source for DEI and PCP.<br>0: Classified PCP and DEI.<br>1: PCP_VAL and DEI_VAL from ES0.<br>2: REW:PORT:PORT_VLAN_CFG.PORT_DEI, REW:PORT:PORT_VLAN_CFG.PORT_PCP.<br>3: DP and QoS mapped to PCP and DEI (per port table). |
| PCP_VAL | 33 | 3 | See QOS_SRC_SEL. |
| DEI_VAL | 36 | 1 | See QOS_SRC_SEL. |
| HIT_STICKY | | 1 | If set, a frame has matched the associated entry. |

## 4.7.5 Range Checkers

The following table lists the registers associated with configuring range checkers.

*Table 50 •* **Range Checker Configuration**

| Register | Description | Replication |
|---|---|---|
| ANA::VCAP_RNG_TYPE_CFG | Configuration of the range checker types | None |
| ANA::VCAP_RNG_VAL_CFG | Configuration of range start and end points | None |

The IS1 entries and the IP4_TCP_UDP entry in IS2 contain eight range checker flags (L4_RNG), which are matched against an 8-bit range key. The range key is generated for each frame based on extracted frame data and the configuration in ANA::VCAP_RNG_TYPE_CFG and ANA::VCAP_RNG_VAL_CFG. Each of the eight range checkers can be configured to one of the following range types:

- TCP/UDP destination port range
  Input to the range is the frame's TCP/UDP destination port number.
- TCP/UDP source port range
  Input to the range is the frame's TCP/UDP source port number.
- TCP/UDP source and destination ports range. Range is matched if either source or destination port is within range.
  Input to the range are the frame's TCP/UDP source and destination port numbers.
- VID range
  IS1: Input to the range is the frame's VID or the port VID if the frame is untagged.
  IS2: Input to the range is the classified VID.
- DSCP range
  IS1: Input to the range is the translated DSCP value from basic classification.
  IS2: Input to the range is the classified DSCP value.

For IS2, the range key is only applicable to TCP/UDP frames. For IS1, the range key is generated for any frame types. Specific range types not applicable to a certain frame type (for example, TCP/UDP port ranges for IPv4 Other frames) must be set to don't care in entries the frame type can match.

Range start points and range end points are configured in ANA::VCAP_RNG_VAL_CFG.

### 4.7.6 VCAP-II Configuration

This section provides information about how the VCAPs IS1, IS2, and ES0 are configured. The following table lists the registers associated with VCAP configuration.

***Table 51 •*** **VCAP Configuration**

| Register | Description | Replication |
|---|---|---|
| VCAP_UPDATE_CTRL | General configuration register | None |
| VCAP_MV_CFG | Move configuration | None |
| VCAP_ENTRY_DAT | Entry data cache | 32 |
| VCAP_MASK_DAT | Entry mask cache | 32 |
| VCAP_ACTION_DAT | Action data cache | 32 |
| VCAP_CNT_DAT | Counter data cache | 32 |
| VCAP_TG_DAT | Type-Group cache | None |
| VCAP_STICKY | Sticky-bit indications | None |

Each VCAP has defined various constants and are accessed using the registers listed in the following table.

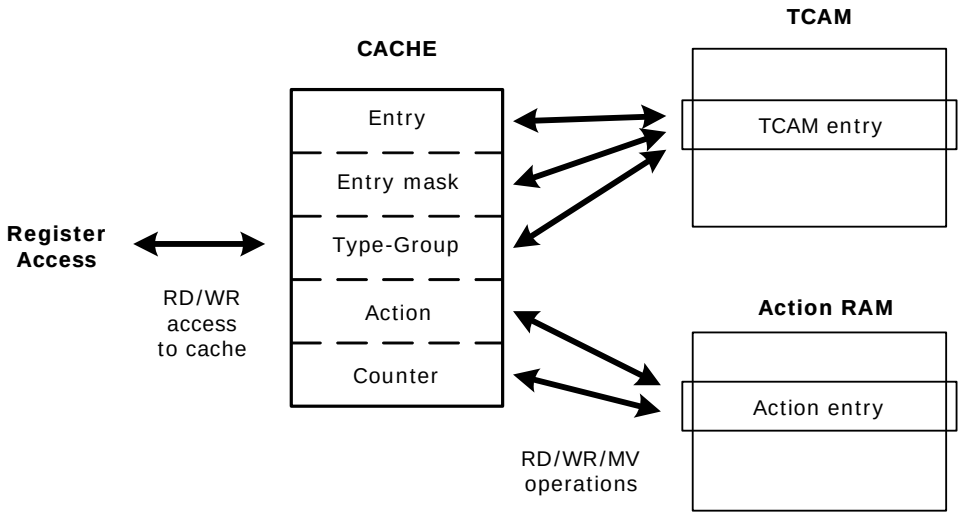
***Table 52 •*** **VCAP Constants**

| Register | Description | Replication |
|---|---|---|
| ENTRY_WIDTH | Width of entry field | None |
| ENTRY_CNT | Number of entries | None |
| ENTRY_SWCNT | Number of subwords | None |
| ENTRY_TG_WIDTH | Width of type-group field | None |
| ACTION_DEF_CNT | Number of default actions | None |
| ACTION_WIDTH | Width of action field | None |
| CNT_WIDTH | Width of counter field | None |

Each VCAP implements its own set of the registers listed in Table 51, page 78 and Table 52, page 78.

Entries in a VCAP are accessed indirectly through an entry and action cache. The cache is accessible using the VCAP configuration registers listed in Table 51, page 78. As shown in the following illustration, an entry in the VCAP consists of a TCAM entry and an associated action and counter entry.

*Figure 23 •* **VCAP Configuration Overview**




A TCAM entry consists of entry data, entry mask, and a type-group value. The type-group value is used internally to differentiate between VCAP lookups of different subword sizes. Each TCAM entry has an associated action entry. Additionally, the action RAM has an entry for each of the default actions in the VCAP. The entries in the action RAM consists of action data and a counter value.

For a write access, the TCAM and action entry must be written to the cache and then copied from the cache to the TCAM/RAM. For a read access, the TCAM and action entry must first be retrieved from the TCAM/RAM before being read from the cache. When a read or write operation is initiated, it is possible to individually select if the operation should be applied to the TCAM and/or action RAM. When data is moved between the cache and the TCAM/RAM, it is always the entire entry that is moved. For VCAPs with several subwords per entry, this must be taken into account if only a single subword of a TCAM entry should be updated. To modify a single subword, the entire TCAM entry must be read, then the subword must be modified in the cache, and finally the entry must be written back to the TCAM.

The cache can hold only one VCAP entry (TCAM and action entry) at a time. After the TCAM and action entry are written to the cache, the cache must be copied to the TCAM and RAM before new entries can be written to the cache.

The following table lists the different parameters for the three VCAPs available in <CHIPID>. The parameters are needed to format the data to be written to the cache. The parameters can also be read in the registers listed in Table 52, page 78.

*Table 53 •* **VCAP Parameters**

| VCAP | Entry Width | Number of Entries | Action Width | Number of Default Actions | Counter Width | Subwords | Type-Group Width |
|---|---|---|---|---|---|---|---|
| IS1 | 188 | 256 | 60 | 0 | 2 (sticky) | 2 | 2 |
| IS2 | 196 | 256 | 46 | 28 | 32 | 1 | 1 |
| ES0 | 29 | 256 | 37 | 26 | 1 (sticky) | 1 | 1 |

#### 4.7.6.1 Creating a VCAP Entry in the Cache

Before a VCAP entry can be created in the TCAM and RAM, the entry must be created in the cache. The cache is accessed through these 32-bit registers:

- VCAP_ENTRY_DAT
- VCAP_MASK_DAT
- VCAP_ACTION_DAT
- VCAP_CNT_DAT

- VCAP_TG_DAT

Each of the cache registers are replicated 32 times, however, only the bits used by the VCAP are mapped to physical registers. For example, for VCAP IS1, only the lowest 188 bits of VCAP_ENTRY_DAT and VCAP_MASK_DAT is mapped to physical registers. As mentioned previously, a VCAP entry consists of a TCAM entry and an action entry.

The TCAM entry consists of entry data, mask data, a type value, and a type-group value. The entry data prefixed with the type value is written to VCAP_ENTRY_DATA. The mask data is written to VCAP_MASK_DATA, and the type-group value is written to VCAP_TG_DAT. The type and type-group values are used internally in the VCAP to distinguish between the different entry types. The following table lists the type and type-group value for each of the entry types.

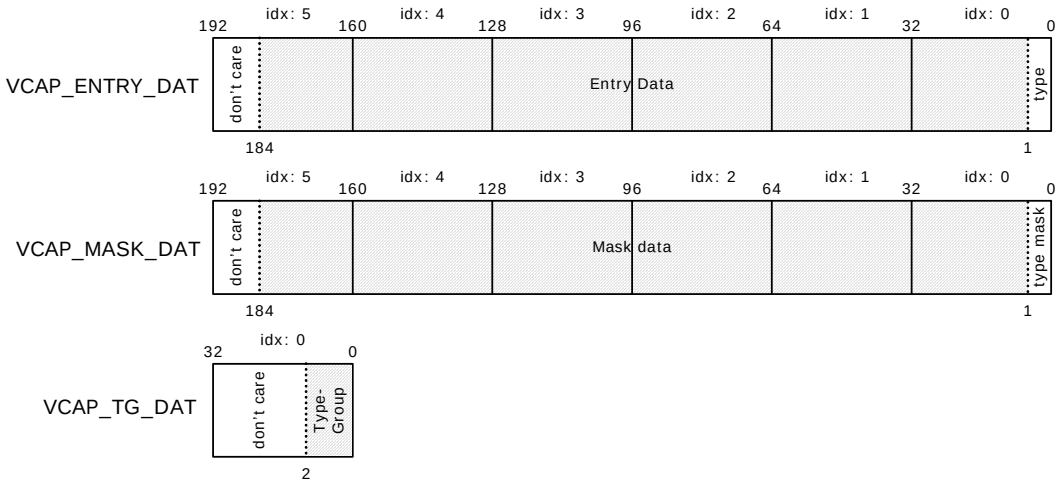
***Table 54 •*** **Entry, Type, and Type-Group Parameters**

| VCAP | Entry Type | Entry Width | Subwords | Type Value [width in ()] | Type-Group Value [width in ()] |
|---|---|---|---|---|---|
| IS1 | IS1 | 183 | 1 | 0 (1) | 1 (2) |
| IS1 | SMAC_SIP4 | 85 | 2 | Not used (0) | 2 (2) |
| IS1 | SMAC_SIP6 | 181 | 1 | 1 (1) | 1 (2) |
| IS2 | MAC_ETYPE | 184 | 1 | 0 (3) | 1 (1) |
| IS2 | MAC_LCC | 184 | 1 | 1 (3) | 1 (1) |
| IS2 | MAC_SNAP | 192 | 1 | 2 (3) | 1 (1) |
| IS2 | ARP | 177 | 1 | 3 (3) | 1 (1) |
| IS2 | IP4_TCP_UCP | 191 | 1 | 4 (3) | 1 (1) |
| IS2 | IP4_OTHER | 190 | 1 | 5 (3) | 1 (1) |
| IS2 | IP6_STD | 192 | 1 | 6 (3) | 1 (1) |
| ES0 | VID | 28 | 1 | Not used (0) | 1 (1) |

Note that the type value is not used for all entry types. If the type value is not used for an entry type, write the entry data from bit 0 of VCAP_ENTRY_DAT.

As an example of how a TCAM entry is laid out in the cache register, the following illustration shows a TCAM entry of the IS1 entry type for the VCAP IS1.

***Figure 24 •*** **Entry Layout In Register Example**

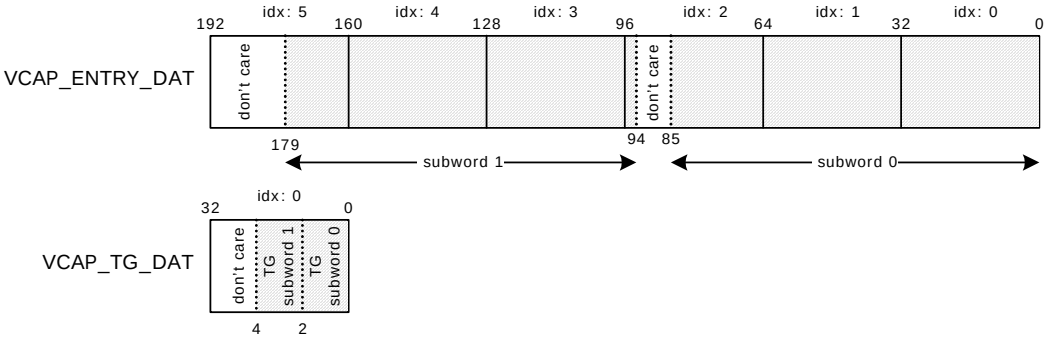



Generally, the type value must never be masked. However, by masking the type bits a lookup in the VCAP is able to match several different entry types. For example, the IS2 entry types MAC_ETYPE and

MAC_LLC have the type values 000 and 001, respectively. By masking bit 0, a lookup is able to match both entry types.

The entry type used in the preceding example only has one subword per entry in the TCAM. Creating a TCAM entry with an entry type that has several subwords per TCAM entry is a little more complicated. In the example shown in the following illustration, the SMAC_SIP4 entry type of the VCAP IS1 is used. The SMAC_SIP4 entry type has two subwords per TCAM entry. From Table 54, page 80, it can be seen that the SMAC_SIP4 entry type has a width of 85 bits per subword. A row in the IS1 TCAM is 188 bits wide For more information, see Table 53, page 79). Each subword is assigned to half a TCAM row; that is, subword 0 is assigned to bits 0-93 and subword 1 is assigned to bits 94-187. Because the SMAC_SIP4 entry only is 85 bits wide, there are nine unused bits for each subword, as shown in the following illustration. Note that the SMAC_SIP4 entry type does not use a type field. The layout for VCAP_MASK_DAT is similar to VCAP_ENTRY_DAT. Additionally, a type-group value is associated to each subword and that the type-group values are laid out back-to-back in VCAP_TG_DAT as shown.

***Figure 25 •*** **Entry Layout In Register Using Subwords Example**




To invalidate an entry in the TCAM (so a lookup never matches the entry), set the type-group for the entry to 0. If there are more subwords in the entry, each subword can be individually invalidated by setting its corresponding type-group value to 0.

The action entry is written to VCAP_ACTION_DAT. Similar to an entry data, an action entry also has a prefixed type value. The following table lists the parameters for the different action types available in VCAPs.

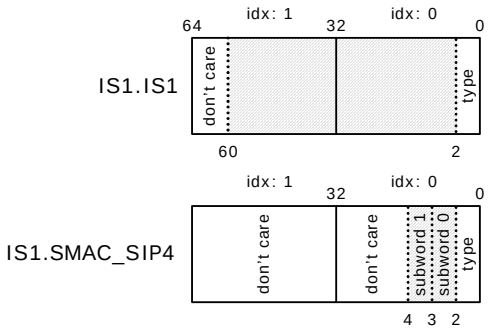
***Table 55 •*** **Action and Type Field Parameters**

| VCAP | Action Type | Action Width | Subwords | Type Value [width in ()] |
|---|---|---|---|---|
| IS1 | IS1 | 58 | 1 | 0 (2) |
| IS1 | SMAC_SIP4 | 1 | 2 | 1 (2) |
| IS1 | SMAC_SIP6 | 1 | 1 | 2 (2) |
| IS2 | BASE_TYPE | 46 | 1 | Not used (0) |
| ES0 | VID | 37 | 1 | Not used (0) |

An action that is associated with an entry type with several subwords per entry has an equal number of subwords. For actions with several subwords, the subwords are simply concatenated together.

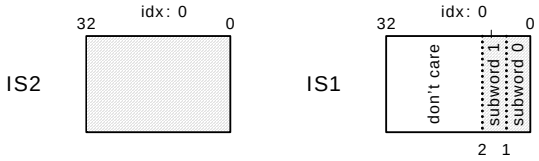
The following illustration shows the action layout in the VCAP_ACTION_DAT register for an IS1 and an SMAC_SIP4 action entry. The IS1 action entry has one subword per row, and the SMAC_SIP4 has two subwords per row.

***Figure 26 •*** **Action Layout in Register Example**




The counter value associated to the action is written to VCAP_CNT_DAT. VCAP_CNT_DAT contains a counter value for each subword in the TCAM entry. For action entries, the counter values for each subword are simply concatenated together. The counter layout for the VCAP_CNT_DAT register the VCAPs IS1 and IS2 is shown in the following illustration. The VCAP IS2 features a 32-bit counter with one subword, and the VCAP IS1 features a 1-bit sticky counter with two subwords.

***Figure 27 •*** **Counter Layout in Register Example**




### 4.7.6.2 Copying Entries Between the Cache and TCAM/RAM

When an entry and associated action is created in the cache, the data in the cache must be copied to a given address in the TCAM and RAM. This is done using the VCAP_UPDATE_CTRL register using the following procedure:

1. Set VCAP_UPDATE_CTRL.UPDATE_CMD to copy from cache to TCAM/RAM.
2. Set the address for the entry in VCAP_UPDATE_CTRL.UPDATE_ADDR.
3. Set VCAP_UPDATE_CTRL.UPDATE_SHOT to initiate the copy operation. The bit is cleared by hardware when the operation is finished.

Initiating another operation before the UPDATE_SHOT field is cleared is not allowed. The delay between setting the UPDATE_SHOT field and the clearing of that field depends on the type of operation and the traffic load on the VCAP.

By setting the fields UPDATE_ENTRY_DIS, UPDATE_ACTION_DIS, and/or UPDATE_CNT_DIS in the VCAP_UPDATE_CTRL register the writing of the TCAM, action, and/or the counter entry can be disabled.

Copying a VCAP entry from the TCAM/RAM to the cache is done in a similar fashion by setting VCAP_UPDATE_CTRL.UPDATE_CMD to copy from TCAM/RAM to the cache. Note that due to internal mapping of the entry data and mask data, the values that are read back from the TCAM cannot always match with the values that were originally written to the TCAM. The internal mapping that happens is listed in the following table. There are differences, because a masked 1 is read back as a masked 0, which functionally is the same.

***Table 56 •*** **Internal Mapping of Entry and Mask**

| Written Entry | Written Mask | Description | Read Entry | Read Mask |
|---|---|---|---|---|
| 0 | 0 | Match-0 | 0 | 0 |
| 0 | 1 | Match-Any | 0 | 1 |
| 1 | 0 | Match-1 | 1 | 0 |
| 1 | 1 | Match-Any | 0 | 1 |

If an entry match is not found during a lookup for a given frame, a default action is selected by the VCAP. Default actions and counter values are copied between the cache and the action RAM similar to a regular VCAP entry. The default actions are stored in the RAM right below the last regular action entry; for example, VCAP IS2 has 256 regular entries, so the first default action in VCAP IS2 is stored at address 256, the second at address 257, and so on. For more information about the number of regular VCAP entries in each VCAP, see Table 53, page 79. When a default action is copied from the cache to the RAM, VCAP_UPDATE_CTRL.UPDATE_ENTRY_DIS must be set to disable the update of the TCAM. If updating of the TCAM is not disabled, the operation may overwrite entries in the TCAM.

The cache can be cleared by setting VCAP_UPDATE_CTRL.CLEAR_CACHE. This sets all replications of VCAP_ENTRY_DAT, VCAP_MASK_DAT, VCAP_ACTION_DAT, VCAP_CNT_DAT, and VCAP_TG_DAT to zeros. The CLEAR_CACHE field is automatically cleared by hardware when the cache is cleared.

## 4.7.7 Advanced VCAP Operations

The VCAP supports a number of advanced operations that allow easy moving and removal of entries and actions during frame traffic.

### 4.7.7.1 Moving Entries and Actions

A number of entries and actions can be moved up by several positions in the TCAM and RAM, and a single entry and action can be moved down by several positions in the TCAM and RAM. This is done using the VCAP_UPDATE_CTRL and VCAP_MV_CFG registers.

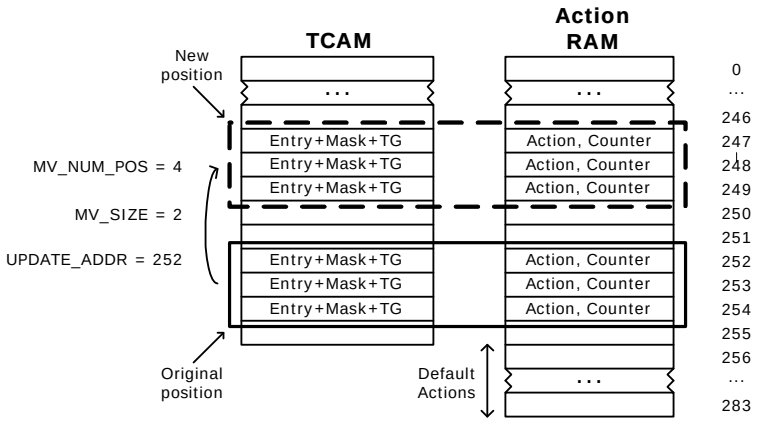
A Move operation is performed by:

- Setting VCAP_UPDATE_CTRL.UPDATE_ADDR equal to the address of the entry with the lowest address, among the entries that must be moved.
- Setting VCAP_MV_CFG.MV_SIZE to the number of entries that must be moved; *n* + 1 entries are moved. Note that a move down operation can only move one entry at a time, which means VCAP_MV_CFG.MV_SIZE must be 0 for move down operations.
- Setting VCAP_MV_CFG.MV_NUM_POS to the number of positions the entries must be moved. The entries are moved *n* positions up or down.
- Setting UPDATE_ENTRY_DIS, UPDATE_ACTION_DIS, and/or UPDATE_CNT_DIS to only move some parts of the VCAP entry.
- Setting VCAP_UPDATE_CTRL.UPDATE_CMD to move up (decreasing addresses) or move down (increasing addresses).
- Initiating the Move operation by setting VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT.

A new command must not be setup until after the VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT field has automatically cleared. Also note that the cache is used by the VCAP while a Move operation is being performed. As a result, any value in cache prior to a Move operation is lost, and a write is not permitted to the cache while a Move operation is performed.

The following illustration shows an example of a Move operation.

***Figure 28 •*** **Move Up Operation Example**



A Move operation can be performed hitlessly during frame traffic, that is, all entries and actions are still available during a Move operation, and all hits are counted by the action hit counters. The TCAM entries at the original positions are invalidated after the Move operation is complete.

During heavy frame traffic, it can take some time for a large move operation to complete, because the moving of individual rows are restarted each time a lookup is performed. If it is not important that the hit counters are accurately updated while the move operation is processed, VCAP_UPDATE_CTRL.MV_TRAFFIC_IGN can be set. This prevents the VCAP from restarting moves and consequently, decreases the time it takes for the move operation to complete. It may, however, lead to inaccurate hit counter values. Note that even if MV_TRAFFIC_IGN is set, the VCAP still processes all lookups correctly.

Default actions can also be moved, however, VCAP_UPDATE_CTRL.UPDATE_ENTRY_DIS must be set.

If a row is moved to a negative address (above address 0), the row is effectively deleted. If a block is partly moved above address 0, the block is also only partially deleted. In other words, the rows that are effectively moved to an address below 0 are not deleted. If one or more rows are deleted during a move operation, the sticky bit VCAP_STICKY.VCAP_ROW_DELETED_STICKY is set.

#### 4.7.7.2 Initializing a Block of Entries

A block of entries can be set to the value of the cache in a single operation. For example, it can be used to initialize all TCAM, action, and counter entries to a specific value. The block of entries to initialize can also include the default action and counter entries.

To perform an initialization operation:

- Set VCAP_UPDATE_CTRL.UPDATE_ADDR equal to the address of the entry with the lowest address, among the entries that should be written.
- Set VCAP_MV_CFG.MV_SIZE to the number of entries that must be included in the initialization operation: *n* + 1 entries are included.
- Set UPDATE_ENTRY_DIS, UPDATE_ACTION_DIS, and/or UPDATE_CNT_DIS to select if the TCAM, action RAM, and/or the counter RAM should be excluded from the initialization operation.
- Set VCAP_UPDATE_CTRL.UPDATE_CMD to the initialization operation.
- Start the initialization operation by setting VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT.

A new command must not be set up until after the VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT field is automatically cleared neither must the cache be written to before VCAP_UPDATE_SHOT is cleared.

## 4.8 Analyzer

The analyzer module is responsible for a number of tasks:

- Determining the set of destination ports, also known as the forwarding decision, for frames received by port modules. This includes Layer-2 forwarding, CPU-forwarding, mirroring, and SFlow sampling.
- Keeping track of network stations and their MAC addresses through MAC address learning and aging.
- Holding VLAN membership information (configured by CPU) and applying this to the forwarding decision.
- Assigning PTP identifiers to PTP frames requesting timestamp updating.

The analyzer consists of three main blocks:

- MAC table
- VLAN table
- Forwarding Engine

The MAC and VLAN tables are the main databases used by the forwarding engine. The forwarding engine determines the forwarding decision and initiates learning in the MAC table when appropriate.

The analyzer operates on analyzer requests initiated by the port modules. For each received frame, the port module requests the analyzer to determine the forwarding decision. Initially, the analyzer request is directed to the VCAP-II. The result from the VCAP-II (the IS2 action) is forwarded to the analyzer along with the original analyzer request. For more information about VCAP-II, see VCAP-II, page 58.

The analyzer request contains the following frame information:

- Destination and source MAC addresses.
- Physical port number where the frame was received (referred to as PPORT).
- Logical port number where the frame was received (referred to as LPORT).
  By default, LPORT and PPORT are the same. However, when using link aggregation, multiple physical ports map to the same logical port. The LPORT value for each physical port is configured in ANA:PORT:PORT_CFG.PORTID_VAL in the analyzer.
- Frame properties derived by the classifier and VCAP-II IS1:
  Classified VID
  Link aggregation code
  Basic CPU forwarding
  CPU forwarding for special frame types determined by the classifier

Based on this information, the analyzer determines an analyzer reply, which is returned to the ingress port modules. The analyzer reply contains:

- The forwarding decision (referred to as DEST). This mask contains 27 bits, 1 bit for each front port and the CPU port.
- The final CPU extraction queue mask (referred to as CPUQ). This mask contains 8 bits, 1 bit for each CPU extraction queue.

The terms PPORT, LPORT, DEST and CPUQ, as previously defined, are used throughout the remainder of this section.

## 4.8.1 MAC Table

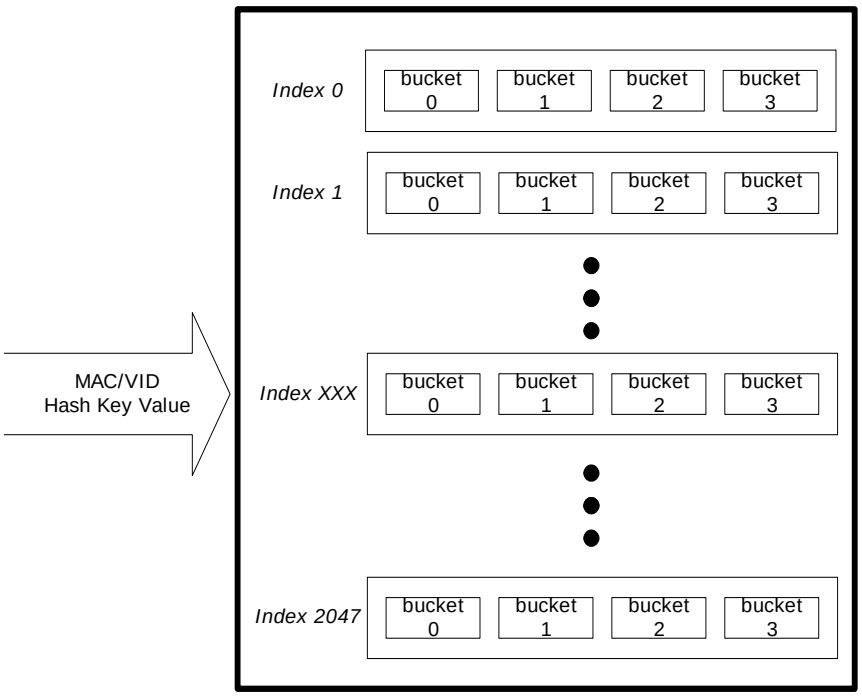
This section provides information about the MAC table block in the analyzer. The following table lists the registers associated with MAC table access.

***Table 57 •*** **MAC Table Access**

| Register | Description | Replication |
|---|---|---|
| MACHDATA | MAC address and VID when accessing the MAC table. | None |
| MACLDATA | MAC address when accessing the MAC table. | None |
| MACTINDX | Direct address into the MAC table for direct read and write. | None |
| MACACCESS | Flags and command when accessing the MAC table. | None |
| MACTOPTIONS | Flags when accessing the MAC table | None |
| AUTOAGE | Age scan period. | None |
| AGENCTRL | Controls the default values for new entries in MAC table. | None |
| ENTRYLIM | Controls limits on number of learned entries per port | Per port |
| LEARNDISC | Counts the number of MAC table entries not learned due lack of storage in the MAC table | None |

The analyzer contains a MAC table with 8192 entries containing information about stations learned by the devices. The table is organized as a hash table with four buckets and 2048 rows. Each row is indexed by an 11-bit hash value, which is calculated based on the station's (MAC, VID) pair, as shown in the following illustration.

***Figure 29 •*** **MAC Table Organization**




The following table lists the fields for each entry in the MAC table.

***Table 58 •*** **MAC Table Entry**

| Field | Bits | Description |
|---|---|---|
| VALID | 1 | Entry is valid. |
| MAC | 48 | The MAC address of the station (primary key). |
| VID | 12 | VLAN identifier that the station is learned with (primary key). |
| DEST_IDX | 6 | Destination mask index pointing to a destination mask in the destination mask table (PGID entries 0 through 63). |
| IP6_MASK | 3 | Partial IPv6 multicast destination port mask. See IPv6 Multicast Entries, page 89. |
| ENTRY_TYPE | 2 | Entry type:<br>0: Normal entry subject to aging.<br>1: Normal entry not subject to aging (locked).<br>2: IPv4 multicast entry not subject to aging. Full port set is encoded in MAC table entry.<br>3: IPv6 multicast entry not subject to aging. Full port set is encoded in MAC table entry. |
| AGED_FLAG | 1 | Entry is aged once by an age scan. See Age Scan, page 87. |
| MAC_CPU_COPY | 1 | Copy frames from or to this station to the CPU. |
| SRC_KILL | 1 | Do not forward frames from this station.<br>**Note** This flag is not used for destination lookups. |
| IGNORE_VLAN | 1 | Do not use the VLAN_PORT_MASK from the VLAN table when forwarding frames to this station. |

Entries in the MAC table can be added, deleted, or updated in three ways:

- Hardware-based learning of source MAC addresses (that is, inserting new (MAC, VID) pairs in the MAC table).
- Age scans (setting AGED_FLAG and deleting entries.)
- CPU commands (for example, for CPU-based learning.)

#### 4.8.1.1 Hardware-Based Learning

The analyzer adds an entry to the MAC table when learning is enabled, and the MAC table does not contain an entry for a received frame's (SMAC, VID). The new entry is formatted as follows:

- VALID is set
- MAC is set to the frame's SMAC
- VID set to the frame's VID
- ENTRY_TYPE is set to 0 (normal entry subject to aging)
- DEST_IDX is set to the frame's LPORT
- MAC_CPU_COPY is set to AGENCTRL.LEARN_CPU_COPY
- SRC_KILL is set to AGENCTRL.LEARN_SRC_KILL
- IGNORE_VLAN is set to AGENCTRL.LEARN_IGNORE_VLAN
- All other fields are cleared

When a frame is received from a known station, that is, the MAC table already contains an entry for the received frame's (SMAC, VID), the analyzer can update the entry as follows.

For entries of entry type 0 (unlocked entries):

- The AGED_FLAG is cleared. This implies the station is active, avoiding the deletion of the entry due to aging.
- If the existing entry's DEST_IDX differs from the frame's LPORT, then the entry's DEST_IDX is set to the frame's LPORT. This implies the station has moved to a new port.

For entries of entry type 1 (locked entries):

- The AGED_FLAG is cleared. This implies the station is active.

Entries of entry types 2 and 3 are never updated, because their multicast MAC addresses are never used as source MAC addresses.

For more information about learning, see SMAC Analysis, page 97.

#### 4.8.1.2 Age Scan

The analyzer scans the MAC table for inactive entries. An age scan is initiated by either a CPU command or automatically performed by the device with a configurable age scan period (AUTOAGE). The age scan checks the flag AGED_FLAG for all entries in the MAC table. If an entry's AGED_FLAG is already set and the entry is of entry type 0, the entry is removed. If the AGED_FLAG is not set, it is set to 1. The flag is cleared when receiving frames from the station identified by the MAC table entry. For more information, see Hardware-Based Learning, page 87.

#### 4.8.1.3 CPU Commands

The following table lists the set of commands that a CPU can use to access the MAC table. The MAC table command is written to MACACCESS.MAC_TABLE_CMD. Some commands require the registers MACLDATA, MACHDATA, and MACTINDX to be preloaded before the command is issued. Some commands return information in MACACCESS, MACLDATA, and MACHDATA.

***Table 59 •*** **MAC Table Commands**

| Command | Purpose | Use |
|---|---|---|
| LEARN | Insert/learn new entry in MAC table. Position given by (MAC, VID) | Configure MAC and VID of the new entry in MACHDATA and MACLDATA. Configure remaining entry fields in MACACCESS. The location in the MAC table is calculated based on (MAC, VID). |
| FORGET | Delete/unlearn entry given by (MAC, VID) | Configure MAC and VID in MACHDATA and MACLDATA. |

*Table 59 •* **MAC Table Commands *(continued)***

| Command | Purpose | Use |
|---|---|---|
| AGE | Start age scan | No preload required. Issue command. |
| READ | Read entry pointed to by (row, column) | Configure row (0-2047) and column (0-3) of the entry to read in:<br>MACTINDX.INDEX (row)<br>MACTINDX.BUCKET (column)<br>MACACCESS.VALID must be 0.<br>When MAC_TABLE_CMD changes to IDLE, MACHDATA, MACLDATA, and MACACCESS contain the information read. |
| LOOKUP | Lookup entry pointed to by (MAC, VID) | Configure MAC and VID of station to look up in MACHDATA and MACLDATA. MACACCESS.VALID must be 1. Issue a READ command. When MAC_TABLE_CMD changes to IDLE, success of the lookup is indicated by MACACCESS.VALID. If successful, MACACCESS contains the entry information. |
| WRITE | Write entry, MAC table position given by (row, column) | Configure MAC and VID of the new entry in MACHDATA and MACLDATA. Configure remaining entry fields in MACACCESS. The location in the MAC table is given by row and column in MACTINDX. |
| INIT | Initialize the table | No preload required. Issue command. |
| GET_NEXT | Get the smallest entry in the MAC table numerically larger than the specified (MAC, VID). The VID and MAC are evaluated as a 60-bit number with the VID being most significant. | Configure MAC and VID of the starting point for the search in MACHDATA and MACLDATA.<br>When MAC_TABLE_CMD changes to IDLE, success of the search is indicated by MACACCESS.VALID. If successful, MACHDATA, MACLDATA, and MACACCESS contain the information read. |
| IDLE | Indicate that MAC table is ready for new command | |

### 4.8.1.4 Known Multicasts

From a CPU, entries can be added to the MAC table with any content. This makes it possible to add a known multicast address with multiple destination ports:

- Set the MAC and VID in MACHDATA and MACLDATA
- Set MACACCESS.ENTRY_TYPE = 1 because this is not an entry subject to aging.
- Set MACACCESS.AGED_FLAG to 0.
- Set MACACCESS.DEST_IDX to an unused value.
- Set the destination mask in the destination mask table pointed to by DEST_IDX to the desired ports.

**Example** All frames in VLAN 12 with MAC address 0x010000112233 are to be forwarded to ports 8, 9, and 12.

This is done by inserting the following entry in the MAC table:

VID = 12
MAC = 0x010000112233
ENTRY_TYPE = 1
VALID = 1

AGED_FLAG = 0
DEST_IDX = 40

and configuring the destination mask table:

PGID[40 = 0x1300.

IPv4 and IPv6 multicast entries can be programmed differently without using the destination mask table. This is described in the following subsection.

### 4.8.1.5 IPv4 Multicast Entries

MAC table entries with the ENTRY_TYPE = 2 settings are interpreted as IPv4 multicast entries.

IPv4 multicasts entries match IPv4 frames, which are classified to the specified VID, and which have DMAC = 0x01005Exxxxxx, where xxxxxx is the lower 24 bits of the MAC address in the entry.

Instead of a lookup in the destination mask table (PGID), the destination set is set to the lower 2 bits of the DEST_IDX value concatenated with the upper 24 bits of the entry MAC address. This is shown in the following table.

***Table 60 •*** **IPv4 Multicast Destination Mask**

| Destination Ports | Record Bit Field |
|---|---|
| Ports 23-0 | MAC[47-24] |
| Ports 25-24 | DEST_IDX[1-0] |

**Example** All IPv4 multicast frames in VLAN 12 with MAC 01005E112233 are to be forwarded to ports 8, 9, and 12. This is done by inserting the following entry in the MAC table entry:

VALID = 1
VID = 12
MAC = 0x001300112233
ENTRY_TYPE = 2
DEST_IDX = 0

### 4.8.1.6 IPv6 Multicast Entries

MAC table entries with the ENTRY_TYPE = 3 settings are interpreted as IPv6 multicast entries:

IPv6 multicasts entries match IPv6 frames, which are classified to the specified VID, and which have DMAC=0x3333xxxxxxxx, where xxxxxxxx is the lower 32 bits of the MAC address in the entry.

Instead of a lookup in the destination mask table (PGID), the destination set is set to AGED_FLAG field concatenated with the IP6_MASK field, the DEST_IDX field and the upper 16 bits the MAC field. This is shown in the following table.

***Table 61 •*** **IPv6 Multicast Destination Mask**

| Destination Ports | Record Bit Field |
|---|---|
| Port 25 | AGED_FLAG |
| Ports 24-22 | IP6_MASK |
| Ports 21-16 | DEST_IDX |
| Ports 15-0 | MAC [47-32] |

**Example** All IPv6 multicast frames in VLAN 12 with MAC 333300112233 are to be forwarded to ports 8, 9, and 12.

This is done by inserting the following entry in the MAC table entry:

VID = 12
MAC = 0x130000112233

ENTRY_TYPE = 3
VALID = 1
AGED_FLAG = 0
IP6_MASK = 0
DEST_IDX = 0

### 4.8.1.7 Port and VLAN Filter

The following table lists the registers associated with the port and VLAN filter.

***Table 62 •*** **VID/Port Filters**

| Register | Description | Replication |
|---|---|---|
| ANAGEFIL | Port and VLAN filter for limiting the target for aging and search operations on MAC table. | None |

The ANAGEFIL register can be used to only hit specific VLANs or ports when doing certain operations. If the filter is enabled, it affects:

- Manual age scan command (MACACCESS.MAC_TABLE_CMD = AGE)
- The LOOKUP and GET_NEXT MAC table commands. For more information, see CPU Commands, page 87.

### 4.8.1.8 Shared VLAN Learning

The following table lists the location of the Filter Identifier (FID) used for shared VLAN learning.

***Table 63 •*** **FID Definition Registers**

| Register | Description | Replication |
|---|---|---|
| IS1_ACTION.FID_SEL | Specifies the use of IS1_ACTION.FID_VAL for the DMAC lookup, the SMAC lookup, or for both lookups. | Per IS1 entry |
| IS1_ACTION.FID_VAL | FID value. | Per IS1 entry |
| AGENCTRL.FID_MASK | Combines multiple VIDs in the MAC table. | None |

In the default configuration, the device is set up to do Independent VLAN Learning (IVL), that is, MAC addresses are learned separately on each VLAN. The device also supports Shared VLAN Learning (SVL), where a MAC table entry is shared among a group of VLANs. For shared VLAN learning, a MAC address and a Filter Identifier (FID) define each MAC table entry. A set of VIDs then map to the FID.

The device supports shared VLAN learning in two ways, either through an IS1 action specifying the FID to use or by using the AGENCTRL.FID_MASK,which controls a mapping between FID and VIDs.

The IS1 action FID_SEL selects whether to use the FID_VAL for the DMAC lookup, for the SMAC lookup, or for both lookups. If set for a lookup, the FID_VAL replaces the VID when calculating the hash key into the MAC table, when comparing with the entry's VID, and when learning. If an IS1 action returns a FID_SEL > 0, it overrules the use of the FID_MASK for the specific lookup.

The 12-bit FID_MASK masks out the corresponding bits in the VID. The FID used for learning and lookup is therefore calculated as FID = VID AND (NOT FID_MASK).

All VIDs mapping to the same FID share the same MAC table entries.

If the FID_MASK is cleared, Independent VLAN Learning is used. This is the default.

**Example** Configure all MAC table entries to be shared among all VLANs.

This is done by setting FID_MASK to 111111111111.

**Example** Split the MAC table into two separate databases: one for even VIDs and one for odd VIDs.

This is done by setting FID_MASK to 111111111110.

#### 4.8.1.9 Learn Limit

The following table lists the registers associated with controlling the number of MAC table entries per port.

***Table 64 •*** **Learn Limit Definition Registers**

| Register | Description | Replication |
|---|---|---|
| ENTRYLIM | Configures maximum number of unlocked entries in the MAC table per ingress port. | Per port |
| PORT_CFG. LIMIT_CPU | If set, learn frames exceeding the limit are copied to the CPU. | Per port |
| PORT_CFG. LIMIT_DROP | If set, learn frames exceeding the limit are discarded. | Per port |
| LEARNDISC | The number of MAC table entries that could not be learned due to a lack of storage space. | None |

The ENTRYLIM.ENTRYLIM register specifies the maximum number of unlocked entries in the MAC table that a port is allowed to use. Locked and IPMC entries are not taken into account.

After the limit is reached, both auto-learning and CPU-based learning on unlocked entries are denied. A learn frame causing the limit to be exceeded can be copied to the CPU (PORT_CFG.LIMIT_DROP) and the forwarding to other front ports can be denied (PORT_CFG.LIMIT_DROP).

The ENTRYLIM.ENTRYSTAT register holds the current number of entries in the MAC table. MAC table aging and manual removing of entries through the CPU cause the current number to be reduced. If a MAC table entry moves from one port to another port, this is also reduces the current number. If the move causes the new port's limit to be exceeded, the entry is denied and removed from the MAC table.

The LEARNDISC counts all events where a MAC table entry is not created or updated due to a learn limit.

### 4.8.2 VLAN Table

The following table lists the registers associated with the VLAN Table.

***Table 65 •*** **VLAN Table Access**

| Register | Description | Replication |
|---|---|---|
| VLANTIDX | VID to access, and VLAN flags. | None |
| VLANACCESS | VLAN port mask for VID and command for access | None |

The analyzer has a VLAN table that contains information about the members of each of the 4096 VLANs. The following table lists fields for each entry in the VLAN table.

***Table 66 •*** **Fields in the VLAN Table**

| Field | Bits | Description |
|---|---|---|
| VLAN_PORT_MASK | 26 | One bit for each port. Set if port is member of VLAN. The CPU port is always a member of all VLANs. |
| VLAN_MIRROR | 1 | Mirror frames received in the VLAN. See Mirroring, page 100. |
| VLAN_SRC_CHK | 1 | VLAN ingress filtering. If set, frames classified to this VLAN are dropped if PPORT is not member of the VLAN. |

*Table 66 •* **Fields in the VLAN Table *(continued)***

| Field | Bits | Description |
|---|---|---|
| VLAN_LEARN_DISABLED | 1 | Disable learning in the VLAN. |
| VLAN_PRIV_VLAN | 1 | Set VLAN to private. |

By default, all ports are members of all VLANs. This default can be changed through a CPU command. The following table lists the set of commands that a CPU can issue to access the VLAN table. The VLAN table command is written to VLANACCESS.VLAN_TBL_CMD.

*Table 67 •* **VLAN Table Commands**

| Command | Purpose | Use |
|---|---|---|
| INIT | Initialize the table | Issue command. When VLAN_TBL_CMD changes to IDLE, initialization has completed and all ports are member of all VLANs. All flags are cleared. |
| READ | Read VLAN table entry for specific VID. | Configure the VLAN to read from in VLANTIDX.INDEX. When VLAN_TBL_CMD changes to IDLE, VLANACCESS and VLANTIDX contain the information read. |
| WRITE | Write VLAN table entry for specific VID. | Configure the VLAN to write to in VLANTIDX.INDEX. Configure the content of the VLAN record in VLANACCESS.VLANACCESS VLANTIDX.VLAN_MIRROR VLANTIDX.VLAN_SRC_CHK VLANTIDX.VLAN_LEARN_DISABLED VLANTIDX.VLAN_PRIV_VLAN |
| IDLE | Indicate that VLAN table is ready for new command | |

## 4.8.3 Forwarding Engine

The analyzer determines the set of ports to which each frame is forwarded, in several configurable steps. The resulting destination port set can include any number of ports, as well as the CPU port.

The analyzer request from the port modules is passed through all the processing steps of the forwarding engine. As each step is carried out, the destination port set (DEST) and CPU extraction queue mask (CPUQ) are built up.

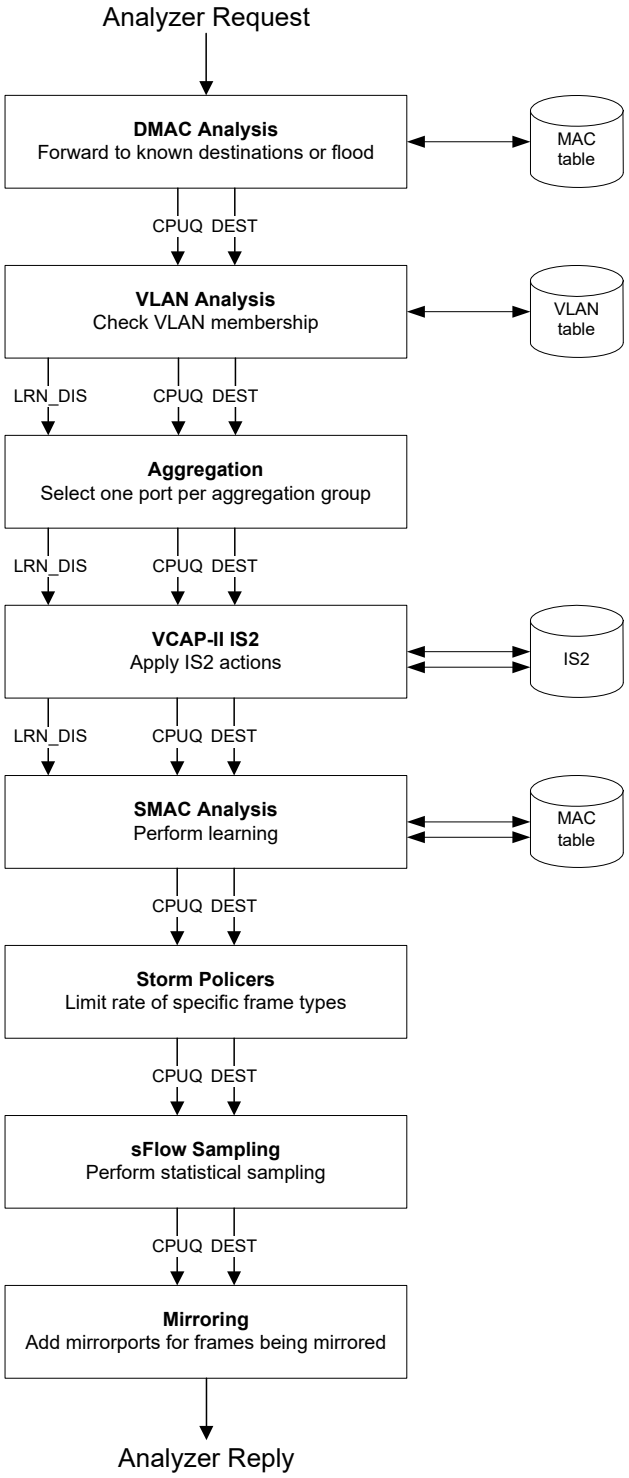
In addition to the forwarding decision, the analyzer determines which frames are subject to learning (also known as learn frames). Learn frames trigger insertion of a new entry in the MAC table or update of an existing entry. Learning is presented as part of the forwarding, because in some cases, learning changes the normal forwarding of a frame, such as secure learning.

During the processing, the analyzer determines a local frame property. The learning-disabled flag, LRN_DIS is used in the SMAC Learning step:

- If the learning-disabled flag is set, learning based on (SMAC, VID) is disabled.
- If the learning-disabled flag is cleared, learning is conducted according to the configuration in the SMAC learning step.

The following illustration shows the configuration steps in the analyzer.

*Figure 30 •* **Analysis Steps**




### 4.8.3.1 DMAC Analysis

During the DMAC analysis step, the (DMAC, VID) pair is looked up in the MAC table to get the first input to the calculation of the destination port set. For more information about the MAC table, see MAC Table, page 85.

The following table lists the registers associated with the DMAC analysis step.

***Table 68 •*** **DMAC Analysis Registers**

| Register | Description | Replication |
|---|---|---|
| FLOODING.FLD_UNICAST | Index into the PGID table used for flooding of unicast frames. | None |
| FLOODING.FLD_BROADCAST | Index into the PGID table used for flooding of broadcast frames. | None |
| FLOODING.FLD_MULTICAST | Index into the PGID table used for flooding of multicast frames, not flooded by the IPMC flood masks. | None |
| FLOODING_IPMC.FLD_MC4_CTRL | Index into the PGID table used for flooding of IPv4 multicast control frames. | None |
| FLOODING_IPMC.FLD_MC4_DATA | Index into the PGID table used for flooding of IPv4 multicast data frames. | None |
| FLOODING_IPMC.FLD_MC6_CTRL | Index into the PGID table used for flooding of IPv6 multicast control frames. | None |
| FLOODING_IPMC.FLD_MC6_DATA | Index into the PGID table used for flooding of IPv6 multicast data frames. | None |
| PGID[63:0] | Destination and flooding masks table | 64 |
| AGENCTRL. IGNORE_DMAC_FLAGS | Controls the use of MAC table flags from (DMAC, VID) entry and flooding flags | None |
| CPUQ_CFG | Configuration of CPU extraction queues | None |

The (DMAC, VID) pair is looked up in the MAC table. If a match is found, the entry is returned and DEST is determined based on the MAC table entry. For more information, see MAC Table, page 85.

If an entry is found in the MAC table entry of ENTRY_TYPE 0 or 1 and the CPU port is set in the PGID pointed to by the MAC table entry, CPU extraction queue PGID.DST_PGID is added to the CPUQ.

If an entry is not found for the (DMAC, VID) in the MAC table, the frame is flooded. The forwarding decision is set to one of the seven flooding masks defined in ANA::FLOODING or ANA::FLOODING_IPMC, based on one of the flood type definitions listed in the following table.

***Table 69 •*** **Forwarding Decisions Based on Flood Type**

| Frame Type | Condition |
|---|---|
| IPv4 multicast data | DMAC = 0x01005E000000 to 0x01005E7FFFFF<br>EtherType = IPv4<br>IP protocol is not IGMP<br>IPv4 DIP outside 224.0.0.x |
| IPv6 multicast data | DMAC = 0x333300000000 to 0x3333FFFFFFFF<br>EtherType = IPv6<br>IPv6 DIP outside 0xFF02::/16 |

*Table 69 •* **Forwarding Decisions Based on Flood Type *(continued)***

| Frame Type | Condition |
|---|---|
| IPv4 multicast control | DMAC = 0x01005E000000 to 0x01005E7FFFFF<br>EtherType = IPv4<br>IP protocol is not IGMP<br>IPv4 DIP inside 224.0.0.x |
| IPv6 multicast control | DMAC = 0x333300000000 to 0x3333FFFFFFFF<br>EtherType = IPv6<br>IPv6 DIP inside 0xFF02::/16 |
| Broadcast | DMAC = 0xFFFFFFFFFFFF<br>non-IPv4-multicast-data<br>non-IPv6-multicast-data<br>non-IPv4-multicast-control<br>non-IPv6-multicast-control |
| Multicast | Bit 40 in DMAC = 1<br>non-broadcast<br>non-IPv4-multicast-data<br>non-IPv6-multicast-data<br>non-IPv4-multicast-control<br>non-IPv6-multicast-control |
| Unicast | Bit 40 in DMAC = 0 |

Additionally, the MAC table flag MAC_CPU_COPY is processed if MAC_CPU_COPY is set, if the CPU port is added to DEST, and if CPUQ_CFG.CPUQ_MAC is added to CPUQ.

The processing of this flag can be disabled through AGENCTRL.IGNORE_DMAC_FLAGS.

Finally, classifier-based CPU-forwarding is processed if:

- The classifier decided to redirect the frame to the CPU, DEST is set to the CPU port only. The corresponding CPU extraction queue is added to CPUQ.
- The classifier decided to copy the frame to the CPU, the CPU port is added to DEST. The corresponding CPU extraction queue is added to CPUQ.

For more information about frame type definitions for CPU forwarding, see Table 29, page 58.

### 4.8.3.2 VLAN Analysis

During the VLAN analysis step, VLAN configuration is taken into account. As a result, ports can be removed from the forwarding decision. For more information about VLAN configuration, see VLAN Table, page 91.

The following table lists the registers associated with VLAN analysis.

*Table 70 •* **VLAN Analysis Registers**

| Register | Description | Replication |
|---|---|---|
| VLANMASK | If PPORT is set in this mask, and PPORT is not member of the VLAN to which the frame is classified, DEST is cleared. This is also called VLAN ingress filtering. | None |
| PORT_CFG.RECV_ENA | If this bit is cleared for PPORT, forwarding from this port to other front ports is disabled, and DEST is cleared. | Per port |

*Table 70 •* **VLAN Analysis Registers *(continued)***

| Register | Description | Replication |
|---|---|---|
| PGID[106:80] | Source port mask. Port mask per port, which specifies allowed destination ports for frames received on PPORT. By default, a port can forward to all other ports except itself. | Per port |
| ISOLATED_PORTS | Private VLAN mask. Isolated ports are cleared in this mask. | None |
| COMMUNITY_PORTS | Private VLAN mask. Community ports are cleared in this mask. | None |
| ADVLEARN.VLAN_CHK | If set and VLAN ingress filtering clears DEST, then SMAC learning is disabled. | None |

The frame's VID is used as an address for lookup in the VLAN table and the returned VLAN information is processed as follows:

- All ports that are not members of the VLAN (VLAN_PORT_MASK) are removed from DEST, except if the (DMAC, VID) match in the MAC table has VLAN_IGNORE set, or if there is no match in the MAC table and AGENCTRL.FLOOD_IGNORE_VLAN is set.
- **Note** These two exceptions are skipped if AGENCTRL.IGNORE_DMAC_FLAGS is set.
- If the VLAN_PRIV_VLAN flag in the VLAN table is set, the VLAN is private, and isolated and community ports must be treated differently. An isolated port is identified as an ingress port for which PPORT is cleared in the ISOLATED_PORTS register. An community port is identified as an ingress port for which PPORT is cleared in the COMMUNITY_PORTS register. For frames received on an isolated port, all isolated and community ports are removed from the forwarding decision. For frames received on a community port, all isolated ports are removed from the forwarding decision.
- If VLAN ingress filtering is enabled, it is checked whether PPORT is member of the VLAN (VLAN_PORT_MASK). If this is not the case, DEST is cleared.

  VLAN ingress filtering is enabled per port in the VLANMASK register or per VLAN in the VLAN_SRC_CHK flag in the VLAN table. If either is set, VLAN ingress filtering is performed.

Next, it is checked whether the ingress port is enabled to forward frames to other front ports and the source mask (PGID[80+PPORT]) is processed as follows:

- If PORT_CFG.RECV_ENA for PPORT is 0, DEST is cleared except for the CPU port.
- Any ports, which are cleared in PGID[80+PPORT], are removed from DEST.

Finally, SMAC learning is disabled by setting the LRN_DIS flag when either of the following two conditions is fulfilled as follows:

- VLAN_LEARN_DISABLED is set in the VLAN table for the VLAN.
- A frame is subject to VLAN ingress filtering (frame dropped due to PPORT not being member of VLAN), and ADVLEARN.VLAN_CHK is set.

### 4.8.3.3 Aggregation

During the aggregation step, link aggregation is handled. The following table lists the registers associated with aggregation.

*Table 71 •* **Analyzer Aggregation Registers**

| Register | Description | Replication |
|---|---|---|
| PGID[79:64] | Aggregation mask table. | 16 |

The purpose of the aggregation step is to ensure that when a frame is destined for an aggregation group, it is forwarded to exactly one of the group's member ports.

For non-aggregated ports, there is a one-to-one correspondence between logical port (LPORT) and physical port (PPORT). The aggregation step does not change the forwarding decision.

For aggregated ports, all physical ports in the aggregation group map to the same logical port, and the entry in the destination mask table for the logical port includes all physical ports, which are members of the aggregation group. As a result, all but one member port must be removed from the destination port set.

The lni aggregation code generated in the classifier is used to look up an aggregation mask in the aggregation masks table. Finally, ports that are cleared in the selected aggregation mask are removed from DEST.

For more information about link aggregation, see Link Aggregation, page 217.

#### 4.8.3.4 VCAP-II Action Handling

During the VCAP IS2 action handling step, the VCAP IS2 actions are processed. The following table lists the processing of the VCAP actions. The order of processing is from top to bottom.

***Table 72 •*** **VCAP IS2 Action Processing**

| IS2 Action Field | Description |
|---|---|
| CPU_COPY_ENA<br>CPU_QU_NUM | If CPU_COPY_ENA is set, the CPU port is added to DEST. The CPU_QU_NUM bit is set in CPUQ. |
| HIT_ME_ONCE<br>CPU_QU_NUM | If HIT_ME_ONCE is set and the HIT_CNT counter is zero, the CPU port is added to DEST. The CPU_QU_NUM bit is set in CPUQ. |
| LRN_DIS | If set, learning is disabled (LRN_DIS flag is set). |
| POLICE_ENA<br>POLICE_IDX | If POLICE_ENA is set (only applies to first lookup), the POLICE_IDX instructs which policer to use for this frame. For more information, see Policers, page 102. |
| MASK_MODE<br>PORT_MASK | The following actions are defined for MASK_MODE.<br>0: No action.<br>1: Permit. Ports cleared in PORT_MASK are removed from DEST.<br>2: Policy. DEST from the DMAC analysis step is replaced with PORT_MASK. The CPU port in DEST is not changed.<br>3: Redirect - DEST as the outcome of the DMAC, VLAN, and Aggregation analysis steps is replaced with PORT_MASK. The CPU port in DEST is not changed. |
| MIRROR_ENA | If MIRROR_ENA is set, mirroring is enabled. This is used in the Mirroring step (see Mirroring, page 100). |
| PTP_ENA | The following actions are defined for PTP_ENA.<br>0: No action.<br>1: Do one-step PTP update.<br>2: Do two-step PTP update.<br>3: Do both one-step and two-step PTP update.<br>See Hardware Timestamping for AVB, page 124. |

### 4.8.3.5 SMAC Analysis

During the SMAC analysis step, the MAC table is searched for a match against the (SMAC, VID), and the MAC table is updated due to learning. The learning part is skipped if the LRN_DIS flag was set by any of the previous steps.

The following table lists the registers associated with SMAC learning.

***Table 73 •*** **SMAC Learning Registers**

| Register | Description | Replication |
|---|---|---|
| PORT_CFG.LEARN_ENA | If set for PPORT, learning is skipped (that is, LEARNAUTO, LEARNCPU, LEARNDROP, LIMIT_CPU, LIMIT_DROP, LOCKED_PORTMOVE_CPU, and LOCKED_PORTMOVE_DROP are ignored). | Per port |
| PORT_CFG.LEARNAUTO | If set for PPORT, hardware-based learning is performed. | Per port |
| PORT_CFG.LEARNCPU | If set for PPORT, learn frames are copied to the CPU. | Per port |
| PORT_CFG.LEARNDROP | If set for PPORT, the CPU drops or forwards learn frames. | Per port |
| PORT_CFG.LIMIT_CPU | If set for PPORT, learn frames for which PPORT exceeds the port's limit are copied to the CPU. | Per port |
| PORT_CFG.LIMIT_DROP | If set for PPORT, learn frames for which PPORT exceeds the port's limit are discarded. | Per port |
| PORT_CFG. LOCKED_PORTMOVE_CPU | If set for PPORT, frames triggering a port move of a locked entry are copied to the CPU. | Per port |
| PORT_CFG. LOCKED_PORTMOVE_DROP | If set for PPORT, frames triggering a port move of a locked entry are discarded. | Per port |
| AGENCTRL.IGNORE_SMAC_FLAGS | Controls the use of the MAC table flags from (SMAC, VID) entry. | None |

Three different type of learn frames are identified:

- **Normal learn frames** Frames for which an entry for the (SMAC, VID) is not found in the MAC table or the (SMAC, VID) entry in the MAC table is unlocked and has a DEST_IDX different from LPORT. In addition, the learn limit for the LPORT must not be exceeded (ENTRYLIM).
- **Learn frames exceeding the learn limit** Same condition as for normal learn frames except that the learn limit for the LPORT is exceeded (ENTRYLIM)
- **Learn frames triggering a port move of a locked MAC table entry** Frames for which the (SMAC, VID) entry in the MAC table is locked and has a DEST_IDX different from LPORT.

For all learn frames, the following must apply before learning related processing is applied:

- Learning is enabled by PORT_CFG.LEARN_ENA.
- The LRN_DIS flag from previous processing steps must be cleared, which implies that:
  – Learning is not disabled due to VLAN ingress filtering
  – Learning is not disabled due to VCAP IS2 action
  – Learning is enabled for the VLAN (VLAN_LEARN_DISABLED is cleared in the VLAN table)

In addition, learning must not be disabled due to the ingress policer having policed the frame. For more information, see Policers, page 102.

If learning is enabled, learn frames are processed according to the setting of the following configuration parameters.

**Normal learn frames:**

- Automatic learning. If PORT_CFG.LEARNAUTO is set for PPORT, the (SMAC, VID) entry is automatically added to the MAC table
- Drop learn frames. If PORT_CFG.LEARNDROP is set for PPORT, DEST is cleared for learn frames. Therefore, learn frames are not forwarded on any ports. This is used for secure learning, where the CPU must verify a station before forwarding is allowed.
- Copy learn frames to the CPU. If PORT_CFG.LEARNCPU is set for PPORT, the CPU port is added to DEST for learn frames and CPUQ_CFG.CPUQ_LRN is set in CPUQ. This is used for CPU based learning.

**Learn frames exceeding the learn limit:**

- Drop learn frames. If PORT_CFG.LIMIT_DROP is set for PPORT, DEST is cleared for learn frames. As a result, learn frames are not forwarded on any ports.
- Copy learn frames to the CPU – If PORT_CFG.LIMIT_CPU is set for PPORT, the CPU port is added to DEST and CPUQ_CFG.CPUQ_LRN is set in CPUQ for learn frames.

**Learn frames triggering a port move of a locked MAC table entry:**

- Drop learn frames. If PORT_CFG.LOCKED_PORTMOVE_DROP is set for PPORT, DEST is cleared for learn frames. Therefore, learn frames are not forwarded on any ports.
- Copy learn frames to the CPU. If PORT_CFG.LOCKED_PORTMOVE_CPU is set for PPORT, the CPU port is added to DEST and CPUQ_CFG.CPUQ_LOCKED_PORTMOVE is added to CPUQ.

Finally, if a match is found in the MAC table for the (SMAC, VID), adjustments can be made to the forwarding decision.

- If the (SMAC, VID) match in the MAC table has SRC_KILL set, DEST is cleared except the CPU port.
- If the (SMAC, VID) match in the MAC table has MAC_CPU_COPY set, the CPU port is added to DEST and CPUQ_CFG.CPUQ_MAC_COPY is added to CPUQ.

The processing of the MAC table flags from the (SMAC, VID) match can be disabled through AGENCTRL.IGNORE_SMAC_FLAGS.

### 4.8.3.6 Storm Policers

The storm policers are activated during the storm policers step. The following table lists the registers associated with storm policers.

***Table 74 •*** **Storm Policer Registers**

| Register | Description | Replication |
|---|---|---|
| STORMLIMIT_CFG | Enable policing of various frame types. | 4 |
| STORMLIMIT_BURST | Configure maximum allowed rates of the different frame types. | None |

The analyzer contains four storm policers that can limit the maximum allowed forwarding frame rate for various frame types. The storm policers are common to all ports and, as a result, measure the sum of traffic forwarded by the switch. A frame can activate several storm policers, and the frame is discarded if any of the activated storm policers exceed a configured rate. The storm policers work independently of other policers in the system (for example, port policers). As a result, frames policed by other policers are still measured by the storm policers.

Each storm policer can be configured to a frame rate ranging from 1 frame per second to 1 million frames per second.

The following table lists the available storm policers.

***Table 75 •*** **Storm Policers**

| Storm Policer | Description |
|---|---|
| Broadcast | Flooded frames with DMAC = 0xFFFFFFFFFFFF. |
| Multicast | Flooded frames with DMAC bit 40 set, except broadcasts. |

*Table 75 •* **Storm Policers *(continued)***

| Storm Policer | Description |
|---|---|
| Unicast | Flooded frames with DMAC bit 40 cleared. |
| Learn | Learn frames copied or redirected to the CPU due to learning (LOCKED_PORTMOVE_CPU, LIMIT_CPU, LEARNCPU). |

For each of the storm policers, a maximum rate is configured in STORMLIMIT_CFG and STORMLIMIT_BURST:

- STORM_UNIT chooses between a base unit of 1 frame per second or 1 kiloframes per second.
- STORM_RATE sets the rate to 1, 2, 4, 8, …, 1024 times the base unit (STORM_UNIT).
- STORM_BURST configures the maximum number of frames in a burst.
- STORM_MODE specifies how the policer affects the forwarding decision. The options are:
  When policing, clear the CPU port in DEST.
  When policing, clear DEST except for the CPU port.
  When policing, clear DEST

Note that frames where the DMAC lookup returned a PGID with the CPU port set are always forwarded to the CPU even when the frame is policed by the storm policers. For more information, see DMAC Analysis, page 93.

### 4.8.3.7 sFlow Sampling

This process step handles sFlow sampling. The following table lists the registers associated with sFlow sampling.

*Table 76 •* **sFlow Sampling Registers**

| Register | Description | Replication |
|---|---|---|
| SFLOW_CFG | Configures sFlow samplers (type and rates). | Per port |
| CPUQ_CFG.CPUQ_SFLOW | CPU extraction queue for sFLow sampled frames. | None |

sFlow is a standard for monitoring high-speed switch networks through statistical sampling of incoming and outgoing frames. Each port in the devices can be setup as an sFlow agent monitoring the particular link and generating sFlow data. If a frame is sFlow sampled, it is copied to the sFlow CPU extraction queue (CPUQ_SFLOW).

An sFlow agent is configured through SFLOW_CFG with the following options:

- SF_RATE specifies the probability that the sampler copies a frame to the CPU. Each frame being candidate for the sampler has the same probability of being sampled. The rate is set in steps of 1/4096.
- SF_SAMPLE_RX enables incoming frames on the port as candidates for the sampler.
- SF_SAMPLE_TX enables outgoing frames on the port as candidates for the sampler.

The Rx and Tx can be enabled independently. If both are enabled, all incoming and outgoing traffic on the port is subject to the statistical sampling given by the rate in SF_RATE.

### 4.8.3.8 Mirroring

This processing step handles mirroring. The following table lists the registers associated with mirroring.

*Table 77 •* **Mirroring Registers**

| Register | Description | Replication |
|---|---|---|
| ADVLEARN.LEARN_MIRROR | For learn frames, ports in this mask (mirror ports) are added to DEST. | None |

*Table 77 •* **Mirroring Registers *(continued)***

| Register | Description | Replication |
|---|---|---|
| AGENCTRL.MIRROR_CPU | Mirror all frames forwarded to the CPU port module | None |
| PORT_CFG.SRC_MIRROR_ENA | Mirror all frames received on an ingress port (ingress port mirroring). | Per port |
| EMIRRORPORTS | Mirror frames that are to be transmitted on any ports set in this mask (egress port mirroring) | None |
| VLANTIDX.VLAN_MIRROR | Mirror all frames classified to a specific VID. | Per VLAN |
| IS2_ACTION.MIRROR_ENA | Mirror when an IS2 action is hit. | Per VCAP IS2 entry |
| MIRRORPORTS | When mirroring a frame, ports in this mask are added to DEST. | None |
| AGENCTRL.CPU_CPU_KILL_ENA | Clear the CPU port if source port is the CPU port and the CPU port is set in DEST. | None |

Frames subject to mirroring are identified based on the following mirror probes:

- Learn mirroring if ADVLEARN.LEARN_MIRROR is set and frame is a learn frame.
- CPU mirroring if AGENCTRL.MIRROR_CPU is set and the CPU port is set in DEST.
- Ingress mirroring if PORT_CFG.SRC_MIRROR_ENA is set.
- Egress mirroring if any port set in EMIRRORPORTS is also set in DEST.
- VLAN mirroring if VLAN_MIRROR set in the VLAN table entry.
- VCAP-II mirroring if an action is hit that requires mirroring.

The following adjustment is made to the forwarding decision for frames subject to mirroring:

- Ports set in MIRRORPORTS are added to DEST.

If the CPU port is set in the MIRRORPORTS, CPU extraction queue CPUQ_CFG.CPUQ_MIRROR is added to the CPUQ.

For learn frames with learning enabled, all ports in ADVLEARN.LEARN_MIRROR are added to DEST. For more information, see SMAC Analysis, page 97.

For more information about mirroring, see Mirroring, page 220.

Finally, if AGENCTRL.CPU_CPU_KILL_ENA is set, the CPU port is removed if the ingress port is the CPU port itself. This is similar to source port filtering done for front ports and prevents the CPU from sending frames back to itself.

## 4.8.4 Analyzer Monitoring

Miscellaneous events in the analyzer can be monitored, which can provide an understanding of the events during the processing steps. The following table lists the registers associated with analyzer monitoring.

*Table 78 •* **Analyzer Monitoring**

| Register | Description | Replication |
|---|---|---|
| ANMOVED | ANMOVED[n] is set when a known station has moved to port n. | None |
| ANEVENTS | Sticky bit register for various events. | None |

*Table 78 •* **Analyzer Monitoring *(continued)***

| Register | Description | Replication |
|---|---|---|
| LEARNDISC | The number of learn events that failed due to a lack of storage space in the MAC table. | None |

Port moves, defined as a known station moving to a new port, are registered in the ANMOVED register. A port move occurs when an existing MAC table entry for (MAC, VID) is updated with new port information (DEST_IDX). Such an event is registered in ANMOVED by setting the bit corresponding to the new port.

Continuously occurring port moves may indicate a loop in the network or a faulty link aggregation configuration.

A list of 27 events, such as frame flooding or policer drop, can be monitored in ANEVENTS.

The LEARNDISC counter registers every time an entry in the MAC table cannot be made or if an entry is removed due to lack of storage.

## 4.9 Policers and Ingress Shapers

Each device has a pool of 256 policers that can be shared between ingress ports, ingress queues, and VCAP IS2 entries. Each ingress port also has an ingress shaper. Both the policers and the shapers can limit the bandwidth of received frames. When configured bandwidth is exceeded, the policers discard frames, while the ingress shaper holds back the traffic in the queue system. Each frame can hit up to three policers and one ingress shaper.

In addition to the policers and ingress shapers described, the devices also support a number of storm policers and an egress scheduler with per-port and per-egress queue shapers. For more information, see Storm Policers, page 99 and Scheduler and Shaper, page 112.

### 4.9.1 Policers

This section explains the functions of the policers. The following table lists the registers associated with policer control.

*Table 79 •* **Policer Control Registers**

| Register | Description | Replication |
|---|---|---|
| ANA:PORT:POL_CFG | Enables use of port and queue policers. | Per port |
| SYS:POL:POL_PIR_CFG | Configures the policer's peak information rate. | 256 |
| SYS:POL:POL_MODE_CFG | Configures the policer's mode of operation. | 256 |
| SYS:POL:POL_PIR_STATE | Current state of the peak information rate bucket. | 256 |
| SYS:PORT:POL_FLOWC | Flow control settings | Per port |
| SYS::POL_HYST | Hysteresis settings. | None |

The pool of policers can be assigned to the following three blocks:

- Ingress ports. Port 'p' use policer 'p'.
- Ingress queues. Ingress queue 'q' on port 'p' use policer 32 + 8x 'p' + 'q'. Each of the eight per-port ingress queues can be assigned to its own policer.
- VCAP IS2. Any remaining policers can be pointed to by IS2_ACTION.POLICE_IDX.

Port and queue policers are enabled through ANA:PORT:POL_CFG.PORT_POL_ENA and ANA:PORT:POL_CFG.QUEUE_POL_ENA. VCAP IS2 policers are enabled by creating IS2 rules with POLICE_ENA and POLICE_IDX actions. IS2 policers actions only apply to the first lookup in IS2.

Each frame can hit a policer from each block; one port policer, one queue policer, and one VCAP IS2 policer. The policers are selected as follows:

- The ingress port where the frame was received points to the port policer.
- The QoS class classified to by the classifier and VCAP IS1 points to the queue policer.
- The POLICE_IDX action from the VCAP IS2 lookup points to the VCAP IS2 policer.

Any frame received by the MAC and forwarded to the classifier is applicable to policing. Frames with errors, pause frames, or MAC control frames are not forwarded by the MAC and, as a result, they are not accounted for in the policers. That is, they are not policed and are not adding to the rate measured by the policers.

In addition, the following special frame types can bypass the policers:

- If ANA:PORT:POL_CFG.POL_CPU_REDIR_8021 is set, frames being redirected to the CPU due to the classifier detecting the frames as being BPDU, ALLBRIDGE, GARP, or CCM/Link trace frames are not policed.
- If ANA:PORT:POL_CFG.POL_CPU_REDIR_IP is set, frames being redirected to the CPU due to the classifier detecting the frames as being IGMP or MLD frames are not policed.

These frames are still considered part of the rates being measured so the frames add to the relevant policer buckets but they are never discarded due to policing.

The order in which the policers are executed is controlled through ANA:PORT:POL_CFG.POL_ORDER. The order can take the following main modes:

- **Serial** The policers are checked one after another. If a policer is closed, the frame is discarded and the subsequent policer buckets are not updated with the frame. The serial order is programmable.
- **Parallel with independent bucket updates** The three policers are working in parallel independently of each other. Each frame is added to a policer bucket if the policer is open, otherwise the frame is discarded. A frame may be added to one policer although another policer is closed.
- **Parallel with dependent bucket updates** The three policers are working in parallel but dependent on each other with respect to bucket updates. A frame is only added to the policer buckets if all three policers are open.

Each of the 256 policers contain a leaky bucket with the following configurations:

- Peak Information Rate (PIR) – Specified in POL_PIR_CFG.PIR_RATE in steps of 100 kbps. Maximum is 3.277 Gbps.
- Peak Burst Size (PBS) – Specified in POL_PIR_CFG.PIR_BURST in steps of 4 kilobytes. Maximum is 252 kilobytes.

Additionally, the following parameters can be configured per policer:

- The leaky bucket calculation can be configured to include or exclude preamble and inter-frame gap through configuration of POL_MODE_CFG.IPG_SIZE.
- Each policer can be configured to measure frame rates instead of bit rates (POL_MODE_CFG.FRM_MODE). The rate unit can be configured to 100 frames per second or 1 frame per second.
- POL_MODE_CFG.OVERSHOOT_ENA controls whether a bucket is allowed to use more than the actual number of tokens in the bucket when accepting a frame (overshooting). If POL_MODE_CFG.OVERSHOOT_ENA is cleared, the number of tokens in the bucket must be larger than the number of tokens required to accept the frame.

By default, a policer discards frames while the policer is closed. A discarded frame is neither forwarded to any ports (including the CPU) nor is it learned.

However, each port policer has the option to run in flow control where the policer instructs the MAC to issue flow control pause frames instead of discarding frames. This is enabled in SYS:PORT:POL_FLOWC. Common for all port policers, POL_HYST.POL_FC_HYST specifies a hysteresis, which controls when the policer can re-open after having closed.

To improve fairness between small and large frames being policed by the same policer, POL_HYST.POL_DROP_HYST specifies a hysteresis, which controls when the policer can re-open after being closed. By setting it to a value larger than the maximum transmission unit, it guarantees that when

the policer opens again, all frames have the same chance of being accepted. This setting only applies to policers working in drop mode.

The current fill level of the leaky buckets can be read in POL_PIR_STATE. The unit is 0.5 bits.

### 4.9.2 Ingress Shapers

The following table lists the registers associated with ingress shaper control.

***Table 80 •*** **Ingress Shaper Control Registers**

| Register | Description | Replication |
|---|---|---|
| SYS:PORT:ISHP_CFG | Configures rate and burst. | Per port |
| SYS:PORT:ISHP_MODE_CFG | Configures mode of operation. | Per port |
| SYS:PORT:ISHP_STATE | Current level of leaky bucket. | Per port |

In addition to the policers, each port has an ingress shaper that controls the rate at which ingress ports are allowed to transfer data to egress ports. An ingress shaper does not discard any frames when its rate is exceeded, but simply holds back the frames in the ingress queues until the rate is below the configured value again. To ensure proper operation of the ingress shapers, all frames on all ports must be assigned the same QoS class when the ingress shapers are enabled.

The ingress shaper is enabled in ISHP_CFG.ISHP_ENA. Each of the ingress shapers contains a leaky bucket with the following configurations:

- Maximum transfer rate is specified in ISHP_CFG.ISHP_RATE in steps of 100 kbps. Maximum is 3.277 Gbps.
- Maximum burst size is specified in ISHP_CFG.ISHP_BURST in steps of 4 kilobytes. Maximum is 252 kilobytes.

Additionally, the following parameters can be configured per ingress shaper:

- The leaky bucket calculation can be configured to include or exclude preamble and inter-frame gap through configuration of ISHP_MODE_CFG.ISHP_IPG_SIZE.
- Each ingress shaper can be configured to measure frame rates instead of bit rates (ISHP_MODE_CFG.ISHP_FRM_MODE). The rate unit can be configured to 100 frames per second or 1 frame per second.

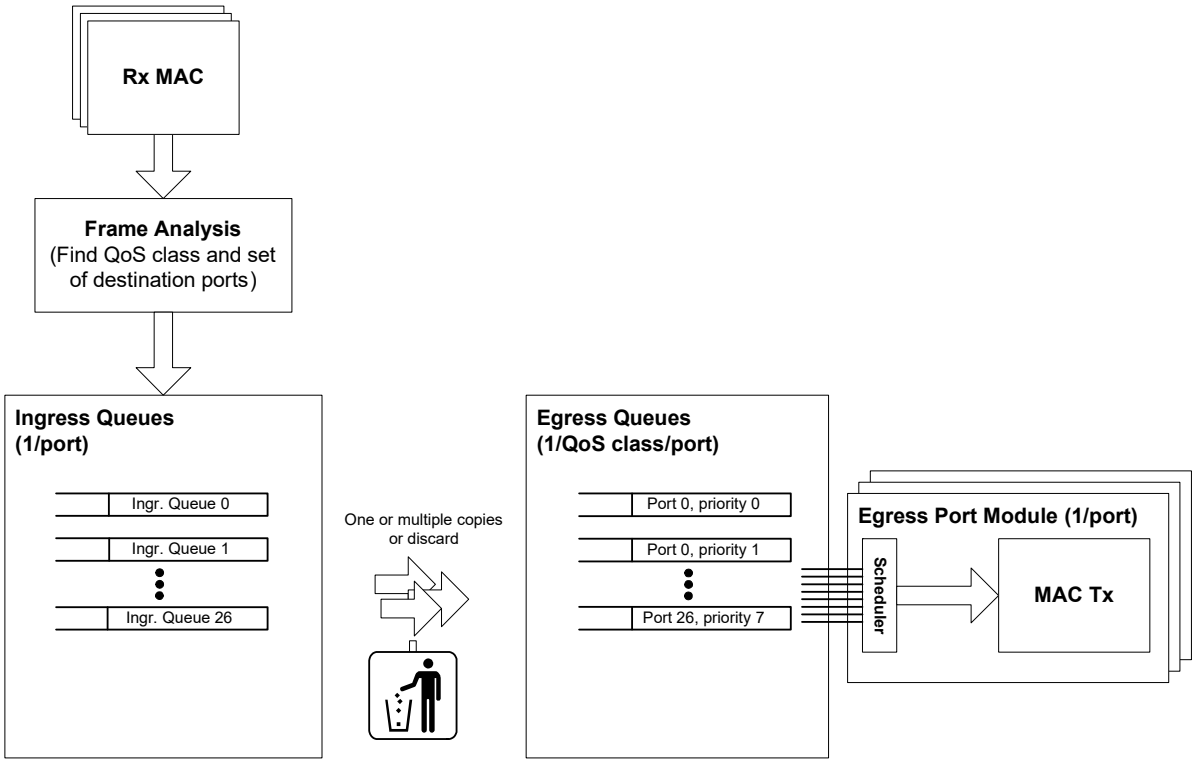
The current fill level of the leaky bucket can be read in ISHP_STATE. The unit is 0.5 bits.

## 4.10 Shared Queue System

The devices include a shared queue system with one ingress queue and eight egress queues per port. The queue system has 512 kilobytes of buffer.

Frames are stored in the ingress queue after frame analysis. Each egress port module selected by the frame analysis receives a copy of the frame and stores the frame in the appropriate egress queue given by the frame's QoS class. The transfer from ingress to egress is extremely efficient with a transfer time of 8 ns per frame copy (equivalent to a transfer rate of 64 Gbps for 64-byte frames and 1.5 Tbps for 1518-byte frames. Each egress port module has a scheduler, which selects between the egress queues when transmitting frames.

The following illustration shows the shared queue system.




Resource depletion can prevent one or more of the frame copies from the ingress queue to the egress queues. If a frame copy cannot be made due to lack of resources, the ingress port's flow control mode determines the behavior as follows:

- Ingress port is in drop mode: The frame copy is discarded.
- Ingress port is in flow control mode: The frame is held back in the ingress queue and the frame copy is made when the congestion clears.

For more information about special configurations of the shared queue system with respect to flow control, see Ingress Pause Request Generation, page 110.

## 4.10.1 Buffer Management

A number of watermarks control how much data can be pending in the egress queues before the resources are depleted. There are no watermarks for the ingress queues, except for flow control, because the ingress queues are empty most of the time due to the fast transfer rates from ingress to egress. For more information, see Ingress Pause Request Generation, page 110. When the watermarks are configured properly, congested traffic does not influence the forwarding of non-congested traffic. F

The memory is split into two main areas:

- A reserved memory area. The reserved memory area is subdivided into areas per port per QoS class per direction (ingress/egress).
- A shared memory area, which is shared by all traffic.

For setting up the reserved areas, egress queue watermarks exist per port and per QoS class for both ingress and egress. The following table lists the reservation watermarks.

***Table 81 •*** **Reservation Watermarks**

| Register | Description | Replication |
|---|---|---|
| BUF_Q_RSRV_E | Configures the reserved amount of egress buffer per egress queue. | Per egress queue |

*Table 81 •* **Reservation Watermarks *(continued)***

| Register | Description | Replication |
|---|---|---|
| BUF_P_RSRV_E | Configures the reserved amount of egress buffer shared among the eight egress queues. | Per egress port |
| BUF_Q_RSRV_I | Configures the reserved amount of egress buffer per ingress port per QoS class across all egress ports. | Per ingress port per QoS class |
| BUF_P_RSRV_I | Configures the reserved amount of egress buffer per ingress port shared among the eight QoS classes. | Per ingress port |

All the watermarks, including the ingress watermarks, are compared against the memory consumptions in the egress queues. For example, the ingress watermarks in BUF_Q_RSRV_I compare against the total consumption of frames across all egress queues received on the specific ingress port and classified to the specific QoS class. The ingress watermarks in BUF_P_RSRV_I compare against the total consumption of all frames across all egress queues received on the specific ingress port.

The reserved areas are guaranteed minimum areas. A frame cannot be discarded or held back in the ingress queues if the frame's reserved areas are not yet used.

The shared memory area is the area left when all the reservations are taken out. The shared memory area is shared between all ports, however, it is possible to configure a set of watermarks per QoS class and per drop precedence level (green/yellow) to stop some traffic flows before others. The following table lists the sharing watermarks.

*Table 82 •* **Sharing Watermarks**

| Register | Description | Replication |
|---|---|---|
| BUF_PRIO_SHR_E | Configures how much of the shared memory area that egress frames with the given QoS class are allowed to use. | Per QoS class |
| BUF_COL_SHR_E | Configures how much of the shared memory area that egress frames with the given drop precedence level are allowed to use. | Per drop precedence level |
| BUF_PRIO_SHR_I | Configures how much of the shared memory area that ingress frames with the given QoS class are allowed to use. | Per QoS class |
| BUF_COL_SHR_I | Configures how much of the shared memory area that ingress frames with the given drop precedence level are allowed to use. | Per drop precedence level |

The sharing watermarks are maximum areas in the shared memory that a given traffic flow can use. They do not guarantee anything.

When a frame is enqueued into the egress queue system, the frame first consumes from the queue's reserved memory area, then from the port's reserved memory area. When all the frame's reserved memory areas are full, it consumes from the shared memory area.

The following provides some simple examples on how to configure the watermarks and how that influences the resource management:

- Setting BUF_Q_RSRV_E(egress port = 17, QoS class = 4) to 2 kilobytes guarantees that traffic destined for port 17 classified to QoS class 4 have room for 2 kilobytes of frame data before frames can get discarded.
- Setting BUF_Q_RSRV_I(ingress port = 17, QoS class = 4) to 2 kilobytes guarantees that traffic received on port 17 classified to QoS class 4 have room for 2 kilobytes of frame data before frames can get discarded.

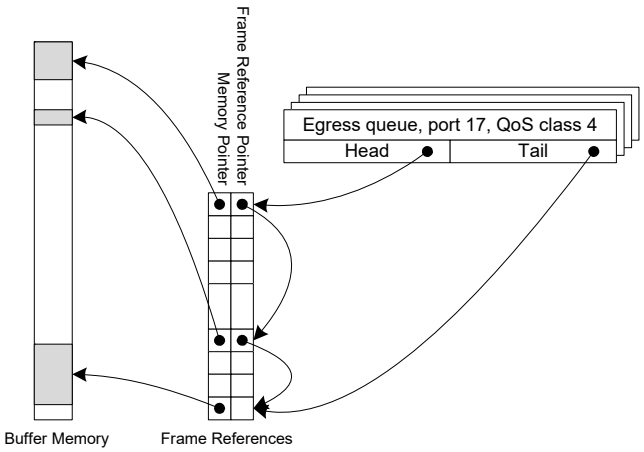
- Setting BUF_P_RSRV_I(ingress port 17) to 10 kilobytes guarantees that traffic received on port 17 have room for 10 kilobytes of data before frames can get discarded.
- The three above reservations reserve in total 14 kilobytes of memory (2 + 2+ 10 kilobytes) for port 17. If the same reservations are made for all ports, there are 512 – 27 × 14 = 134 kilobytes left for sharing. If the sharing watermarks are all set to 134 kilobytes, all traffic groups can consume memory from the shared memory area without restrictions.

  If, instead, setting BUF_PRIO_SHR_E(QoS class = 7) to 100 kilobytes and the other watermarks BUF_PRIO_SHR_E(QoS class = 0:6) to 70 kilobytes guarantees that traffic classified to QoS class 7 has 30 kilobytes extra buffer. The buffer is shared between all ports.

## 4.10.2 Frame Reference Management

Each frame in an egress queue consumes a frame reference, which is a pointer element that points to the frame's data in the memory and to the pointer element belonging to the next frame in the queue. The following illustrations shows how the frame references are used for creating the queue structure.

***Figure 31 •*** **Frame Reference**




The shared queue system holds a table of 5500 frame references. The consumption of frame references is controlled through a set of watermarks. The set of watermarks is the exact same as for the buffer control. The frame reference watermarks are prefixed REF_. Instead of controlling the amount of consumed memory, they control the number of frame references. Both reservation and sharing watermarks are available. For more information, see Table 81, page 105 and Table 82, page 106.

When a frame is enqueued into the shared queue system, the frame consumes first from the queue's reserved frame reference area, then from the port's reserved frame reference area. When all the frame's reserved frame reference areas are full, it consumes from the shared frame reference area.

## 4.10.3 Resource Depletion Condition

A frame copy is made from an ingress port to an egress port when both a memory check and a frame reference check succeed. The memory check succeeds when at least one of the following conditions is met:

- Ingress memory is available: BUF_Q_RSRV_I or BUF_P_RSRV_I are not exceeded.
- Egress memory is available: BUF_Q_RSRV_E or BUF_P_RSRV_E are not exceeded.
- Shared memory is available: None of BUF_PRIO_SHR_E, BUF_COL_SHR_E, BUF_PRIO_SHR_I, or BUF_COL_SHR_I are exceeded.

The frame reference check succeeds when at least one of the following conditions is met:

- Ingress frame references are available: REF_Q_RSRV_I or REF_P_RSRV_I are not exceeded.
- Egress frame references are available: REF_Q_RSRV_E or REF_P_RSRV_E are not exceeded.
- Shared frame references are available: None of REF_PRIO_SHR_E, REF_COL_SHR_E, REF_PRIO_SHR_I, or REF_COL_SHR_I are exceeded.

## 4.10.4 Configuration Example

This section provides an example of how the watermarks can be configured for a QoS-aware switch with no color handling and the effects of the settings.

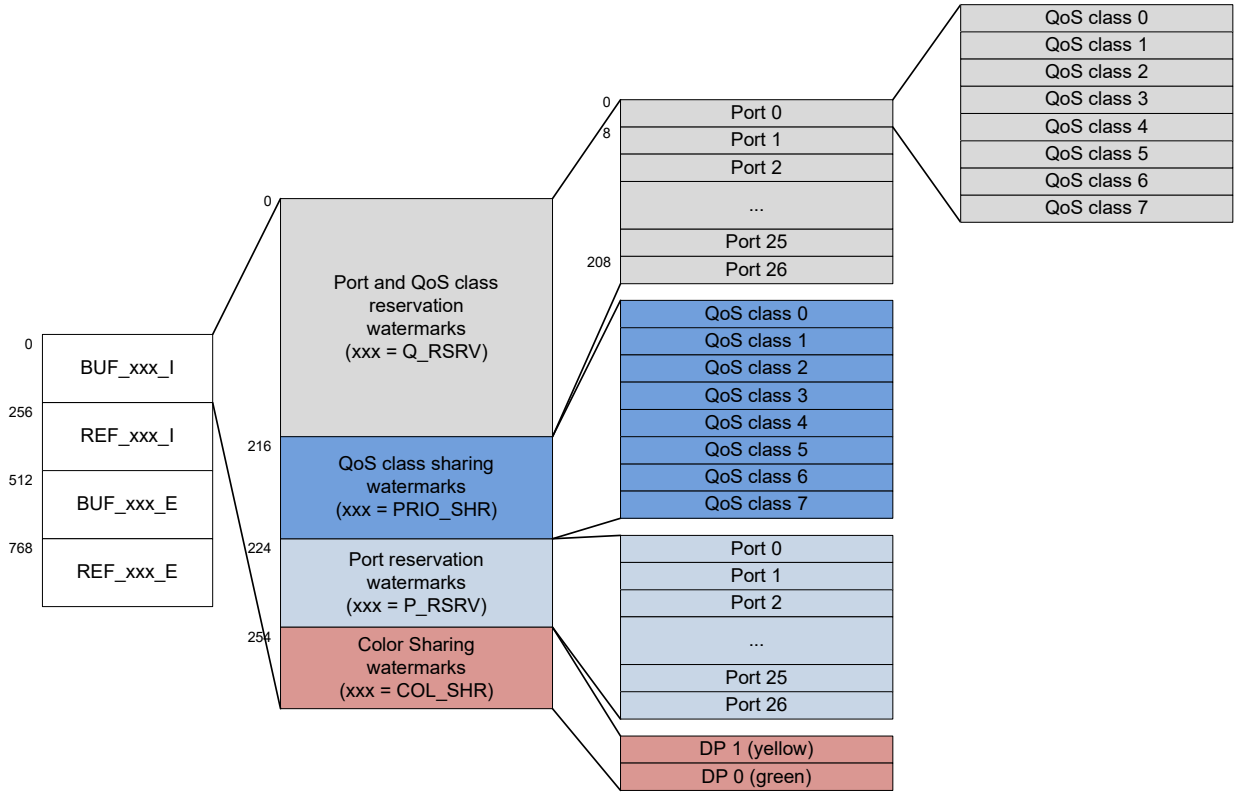
***Table 83 •*** **Watermark Configuration Example**

| Watermark | Value | Comment |
|---|---|---|
| BUF_Q_RSRV_I | 500 bytes | Guarantees that a port is capable of receiving at least one frame in all QoS classes.<br>**Note** It is not necessary to assign a full MTU, because the watermarks are checked before the frame is added to the memory consumption. |
| BUF_P_RSRV_I | 0 | No additional guarantees for the ingress port. |
| BUF_Q_RSRV_E | 200 bytes | Guarantees that all QoS classes are capable of sending a non-congested stream of traffic through the switch. |
| BUF_P_RSRV_E | 10 kilobytes | Guarantees that all egress ports have 10 kilobytes of buffer, independently of other traffic in the switch. This is the most demanding reservation in this setup, reserving 270 kilobytes of the total 512 kilobytes. |
| BUF_COL_SHR_E<br>BUF_COL_SHR_I | Maximum | Effectively disables frame coloring as watermark is never reached. |
| BUF_PRIO_SHR_E<br>BUF_PRIO_SHR_I | 82 kilobytes to 103 kilobytes | The different QoS classes are cut-off with 3 kilobytes distance (82, 85, 88, 91, 94, 97, 100, and 103 kilobytes). This gives frames with higher QoS classes a larger part of the shared buffer area. Effectively, this means that the burst capacity is 92 kilobytes for frames belonging to QoS class 0 and up to 113 kilobytes for frame belonging to QoS class 7. |
| REF_Q_RSRV_E<br>REF_Q_RSRV_I | 4 | For both ingress and egress, this guarantees that four frames can be pending from and to each port. |
| REF_P_RSRV_E<br>REF_P_RSRV_I | 20 | For both ingress and egress, this guarantees that an extra 20 frames can be pending, shared between all QoS classes within the port. |
| REF_COL_SHR_E<br>REF_COL_SHR_I | Maximum | Effectively disables frame coloring as watermark is never reached. |
| REF_PRIO_SHR_E<br>REF_PRIO_SHR_I | 2350 - 2700 | The different QoS classes are cut-off with a distance of 50 frame references (2350, 2400, 2450, 2500, 2550, 2600, 2650, and 2700). This gives frames with higher QoS classes a larger part of the shared reference area. |

## 4.10.5 Watermark Programming and Consumption Monitoring

The watermarks previously described are all found in the SYS::RES_CFG register. The register is replicated 1024 times. The following illustration the organization.

***Figure 32 •*** **Watermark Layout**




The illustration shows the watermarks available for the BUF_xxx_I group of watermarks. For the other groups of watermarks (BUF_xxx_I, REF_xxx_I, BUF_xxx_E, and REF_xxx_E), the exact same set of watermarks is available.

For monitoring purposes, SYS::RES_STAT provides information about the resource consumption currently in use as well as the maximum consumption for corresponding watermarks. The information is available for each of the watermarks listed, and the layout of the RES_STAT register follows the layout of the watermarks. SYS::MMGT.FREECNT holds the amount of free memory in the shared queue system and SYS::EQ_CTRL.FP_FREE_CNT holds the number of free frame references in the shared queue system.

## 4.10.6 Advanced Resource Management

A number of additional handles into the resource management system are available for special use of the device. They are described in the following table.

***Table 84 •*** **Resource Management**

| Resource Management | Description |
|---|---|
| Forced drop of egress frames | SYS:PORT:EGR_DROP_FORCE.<br>If an ingress port is in configured in flow control mode, frames received on the port are by default held back if one or more destination ports do not allow more data. However, if forced drop of egress frames is enabled for the egress port, frames are discarded. This could be enabled for the CPU port and for a mirror target port in order not to cause head-of-line blocking of non-congested traffic. |

*Table 84 •* **Resource Management** ***(continued)***

| Resource Management | Description |
|---|---|
| Prevent ingress port from using of the shared resources. | SYS:IGR_NO_SHARING.<br>For frames received on ports set in this mask, the shared watermarks are considered exceeded. This prevents the port from using more resources than allowed by the reservation watermarks. |
| Prevent egress port from using of the shared resources. | SYS:EGR_NO_SHARING.<br>For frames switched to ports set in this mask the shared watermarks are considered exceeded. This prevents the port from using more resources than allowed by the reservation watermarks. |
| Preferred sources | SYS::EQ_PREFER_SRC.<br>By default, ingress ports that have frames for transmission of equal QoS class are serviced in round robin. However, ingress ports marked in this mask are preferred over ingress ports not marked. |
| Truncating | SYS:PORT:EQ_TRUNCATE.<br>Each egress queue can be configured to truncate frames to 92 bytes. Frames shorter than 92 bytes are not changed. This could be the enabled for a specific CPU extraction queue used for learning or a mirror target port where the first segment of the frames is sufficient for further frame processing. |
| Prevent dequeuing | SYS:PORT:PORT_MODE.DEQUEUE_DIS.<br>Each egress port can disable dequeuing of frames from the egress queues. |

## 4.10.7 Ingress Pause Request Generation

During resource depletion, the shared queue system either discards frames when the ingress port operates in drop mode, or holds back frames when the ingress port operates in flow control mode. The following describes special configuration for the flow control mode.

The shared queue system is enabled for holding back frames during resource depletion in SYS:PORT:PAUSE_CFG.PAUSE_ENA. In addition, this enables the generation of pause requests to the port module based on memory consumptions. The MAC uses the pause request to generate pause frames or create back pressure collisions to halt the link partner. This is done according to the MAC configuration. For more information about MAC configuration, see MAC, page 16.

The shared queue system generates the pause request based on the ingress port's memory consumption and also based on the total memory consumption in the shared queue system. This enables a larger burst capacity for a port operating in flow control while not jeopardizing the non-dropping flow control.

Generating the pause request partially depends on a memory consumption flag, TOT_PAUSE, which is set and cleared under the following conditions:

- The TOT_PAUSE flag is set when the total consumed memory in the shared queue system exceeds the SYS:PORT:PAUSE_TOT_CFG.PAUSE_TOT_START watermark.
- The TOT_PAUSE flag is cleared when the total consumed memory in the shared queue system is below the SYS:PORT:PAUSE_TOT_CFG.PAUSE_TOT_STOP watermark.

The pause request is asserted when both of the following conditions are met:

- The TOT_PAUSE flag is set.
- The ingress port memory consumption exceeds the SYS:PORT:PAUSE_CFG.PAUSE_START watermark.

The pause request is deasserted the following condition is met:

- The ingress port's consumption is below the SYS:PORT:PAUSE_CFG.PAUSE_STOP watermark.

## 4.10.8 Tail Dropping

The shared queue system implements a tail dropping mechanism where incoming frames are discarded if the port's memory consumption and the total memory consumption exceed certain watermarks. Tail dropping implies that the frame is discarded unconditionally. All ports in the device are subject to tail dropping. It is independent of whether the port is in flow control mode or drop mode.

Tail dropping can be effective under special conditions. For example, tail dropping can prevent an ingress port from consuming all the shared memory when pause frames are lost or the link partner is not responding to pause frames.

The shared queue system initiates tail dropping by discarding the incoming frame if the following two conditions are met at any point while writing the frame data to the memory:

- The ingress port memory consumption exceeds the SYS:PORT:ATOP_CFG.ATOP watermark.
- The total consumed memory in the shared queue system exceeds the SYS:PORT:ATOP_TOT_CFG.ATOP_TOT watermark.

## 4.10.9 Test Utilities

This section describes some of test utilities that are built into the shared queue system.

Each egress port can enable a frame repeater (SYS::REPEATER), which means that the head-of-line frames in the egress queues are transmitted but not dequeued after transmission. As a result, the scheduler sees the same frames again and again while the repeater function is active.

The SYS:PORT:PORT_MODE.DEQUEUE_DIS disables both transmission and dequeuing from the egress queues when set.

## 4.10.10 Energy Efficient Ethernet

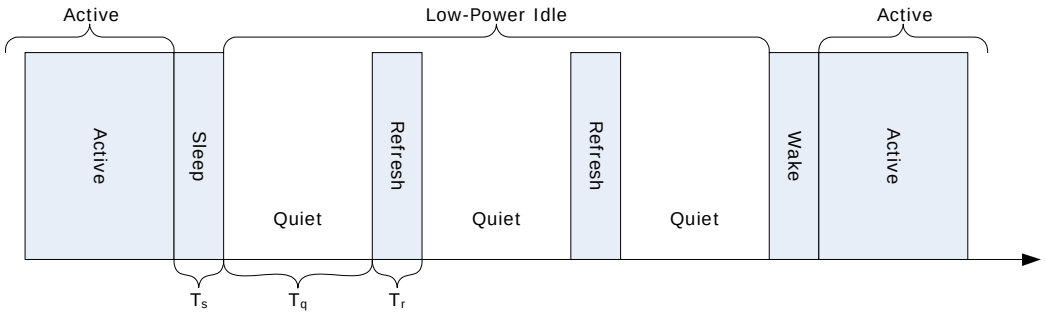
This section provides information about the functions of Energy Efficient Ethernet in the shared queue system. The following tables lists the registers associated with Energy Efficient Ethernet.

***Table 85 •*** **Energy Efficient Ethernet Control Registers**

| Register | Description | Replication |
|---|---|---|
| SYS:PORT:EEE_CFG | Enabling and configuration of Energy Efficient Ethernet | Per port |
| SYS:EEE_THRES | Configuration of thresholds (bytes and frames) | None |
| SYS::SW_STATUS.PORT_LPI | Status bit indicating that egress port is in LPI state | Per port |

The shared queue system supports Energy Efficient Ethernet (EEE) as defined by IEEE 802.3az by initiating the Low Power Idle (LPI) mode during periods of low link use. EEE is controlled per port by an egress queue state machine that monitors the queue fillings and ensures correct wake-up and sleep timing. The egress queue state machine is responsible for informing the connected PCS or internal PHY of changes in EEE states (active, sleep, low power idle, and wake up).

***Figure 33 •*** **Low Power Idle Operation**




Energy Efficient Ethernet is enabled per port through SYS:PORT:EEE_CFG.EEE_ENA.

By default, the egress port is transmitting enqueued data. This is the active state. If none of the port's egress queues have enqueued data for the time specified in SYS:PORT:EEE_CFG.EEE_TIMER_HOLDOFF, the egress port instructs the PCS or internal PHY to enter the EEE sleep state.

When data is enqueued in any of the port's egress queues, a timer (SYS:PORT:EEE_CFG.EEE_TIMER_AGE) is started. When one of the following conditions is met, the port enters the wake up state:

- A queue specified as high priority (SYS:PORT:EEE_CFG.EEE_FAST_QUEUES) has any data to transmit.
- The total number of frames in the port's egress queues exceeds SYS::EEE_THRESS.EEE_HIGH_FRAMES.
- The total number of bytes in the port's egress queues exceeds SYS::EEE_THRESS.EEE_HIGH_FRAMES.
- The time specified in SYS:PORT:EEE_CFG.EEE_TIMER_AGE has passed.

PCS and or the internal PHY is instructed to wake up. To ensure that PCS, PHY, and link partner are resynchronized; the egress port holds back transmission of data until the time specified in SYS:PORT:EEE_CFG.EEE_TIMER_WAKEUP has passed. After this time interval, the port resumes transmission of data.

The status bit SYS::SW_STATUS.PORT_LPI is set while the egress port holds back data due to LPI (from the sleep state to the wake up state, both included).

## 4.11 Scheduler and Shaper

The following table lists the registers associated with the scheduler and egress shaper control.

***Table 86 •*** **Scheduler and Egress Shaper Control Registers**

| Register | Description | Replication |
|---|---|---|
| SYS::LB_DWRR_FRM_ADJ | Configuration of gap value | Common |
| SYS::LB_DWRR_CFG | Enabling of gap value adjustment for use in scheduler and shapers | Per port |
| SYS::SCH_DWRR_CFG | Enabling of DWRR scheduler and configurations of costs | Per port |
| SYS::SCH_SHAPING_CTRL | Enabling of shaping | Per port |
| SYS::SCH_LB_CTRL.LB_INIT | Initialization of scheduler and shapers | Common |
| SYS::LB_THRES | Configuration of shaper threshold | Per shaper |
| SYS::LB_RATE | Configuration of shaper rate | Per shaper |

Each egress port contains a scheduler and a set of egress shapers that control the read out from the egress queuing system to the associated port module.

By default, the scheduler operates in strict priority. The egress queues are searched in the following prioritized order: Queue for QoS class 7 has highest priority followed by 6, 5, 4, 3, 2, 1, and 0.

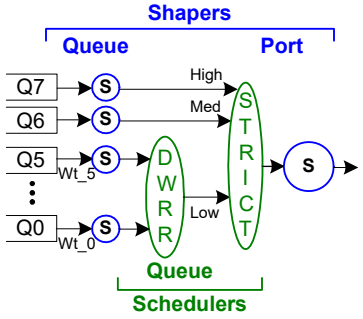
In addition, the scheduler can operate in a mixed mode, where queue 7 and queue 6 are strictly served and queues 5 through 0 operate in a deficit weighted round robin (DWRR) mode. In DWRR mode, QoS class queues 5 through 0 are given a weight and the scheduler selects frames from these queues according to the weights.

Both the egress port and each of the egress queues have an associated leaky-bucket shaper. The egress port shaper is positioned towards the MAC and limits the overall transmission bandwidth on the port. Frames are only scheduled if the port shaper is open. The egress queue shapers control the input to the scheduler for each egress queue. Generally, the scheduler only searches an egress queue if the egress queue's shaper is open.

DWRR is used to guarantee queues a minimum share of the available bandwidth, and shaping is used to configure a maximum rate that cannot be exceeded.

The following illustration shows the egress shapers and scheduler.

***Figure 34 •*** **Egress Scheduler and Shapers**




The overall scheduling algorithm is as follows:

1. If the port shaper is closed, no frames are scheduled. Frames are held back until the port shaper opens.
2. If the port shaper is open, queues with an open queue shaper are candidates for scheduling. Queue 7 has highest priority followed by 6. Queues 5 through 0 may operate in strict mode or in the DWRR mode where each queue is weighted relatively to the other queues. Frames in a queue with a closed queue shaper are held back until the queue shaper opens.
3. If no frames are scheduled during step 2, a second round of scheduling is performed. Queues programmed as work conserving and having a closed queue shaper become candidates for the second round of scheduling.

The following are the configuration options for the shapers and scheduler. Each port is configured independently of other ports. Within a port, the following functionality can be enabled independently:

- DWRR mode (SCH_DWRR_CFG.DWRR_MODE): If set, queues 5 through 0 are scheduled according to the associated weights.
- Port shaping (SCH_SHAPING_CTRL.PORT_SHAPING_ENA): If set, the egress bandwidth is controlled by the port shaper settings.
- Per-queue shaping (SCH_SHAPING_CTRL.PRIO_SHAPING_ENA): If set for a queue, the queue shaper settings control the rate into the scheduler.

## 4.11.1 Egress Shapers

Each of the egress shapers (port and queues) contains a leaky bucket with the following configurations:

- Maximum rate – Specified in LB_RATE.LB_RATE in steps of 100160 bps. Maximum is 3.282 Gbps.
- Maximum burst size – Specified in LB_THRES.LB_THRES in steps of 4 kilobytes. Maximum is 252 kilobytes.

The frame adjustment value LB_DWRR_FRM_ADJ.FRM_ADJ can be used to program the fixed number of extra bytes to add to each frame transmitted (irrespective of QoS class) in the shaper and DWRR

calculations. A value of 20 bytes corresponds to line-rate calculation and accommodates for 12 bytes of inter-frame gap and 8 bytes of preamble. Data-rate based shaping and DWRR calculations are achieved by programming 0 bytes.

Each port can enable the use of the frame adjustment value LB_DWRR_FRM_ADJ.FRM_ADJ through LB_DWRR_CFG.FRM_ADJ_ENA. If enabled on a port, both shapers and scheduler are affected.

By default, while a queue shaper is closed, frames in the queue are not scheduled, even if none of the other queues have frames to transmit. Each queue can enable a work-conserving mode (SCH_SHAPING_CTRL.PRIO_LB_EXS_ENA) in which a second scheduling round is possible. If none of the queues with an open shaper have frames for transmission, work-conserving queues with closed shapers may get a share of the excess bandwidth. The sharing of the excess bandwidth obeys the same configured scheduling rules as for the first round of scheduling.

The queue shapers implement two burst modes. By default, a leaky bucket is continuously assigned new credit according to the configured shaper rate (LB_RATE). This implies that during idle periods, credit is building up, which allows for a burst of data when the queue again has data to transmit. This is not convenient in an Audio/Video Bridging (AVB) environment where this behavior enforces a requirement for larger buffers in end-equipment. To circumvent this, each queue shaper can enable an AVB mode (SCH_SHAPING_CTRLPRIO_LB_AVB_ENA) in which credit is only assigned during periods where the queue shaper has data to transmit and is waiting for another queue to finish a transmission. This AVB mode prevents the accumulation of large amount of credits.

The shapers must be initialized through SCH_LB_CTRL.LB_INIT before use.

### 4.11.2 Deficit Weighted Round Robin

The DWRR uses a cost-based algorithm compared to a weight-based algorithm. A high cost implies a small share of the bandwidth. When the DWRR is enabled, each of queues 5 through 0 are programmed with a cost (SCH_DWRR_CFG.COST_CFG). A cost is a number between 1 and 32.

The programmable DWRR costs determine the behavior of the DWRR algorithm. The costs result in weights for each queue. The weights are relative to one another, and the resulting share of the egress bandwidth for a particular QoS class is equal to the queue's weight divided by the sum of all the queues' weights.

Costs are easily converted to weights and vice versa given the following two algorithms:

**Weights to Costs** Given a desired set of weights (W0, W1, W2, W3, W4, W5), the costs can be calculated using the following algorithm:

1. Set the cost of the queue with the smallest weight (Wsmallest) to cost 32.
2. For any other queue Qn with weight Wn, set the corresponding cost Cn to:
   Cn = 32 × Wsmallest/Wn

**Costs to Weights** Given a set of costs for all queues (C0, C1, C2, C3, C4, C5), the resulting weights can be calculated using the following algorithm:

1. Set the weight of the queue with the highest cost (Chighest) to 1.
2. For any other queue Qn with cost Cn, set the corresponding weight Wn to Wn = Chighest/Cn

**Cost and Weight Conversion Examples**

The following bandwidth distribution must be implemented:

- Queue 0: 5% (W0 = 5)
- Queue 1: 10% (W1 = 10)
- Queue 2: 15% (W2 = 15)
- Queue 3: 20% (W3 = 20)
- Queue 4: 20% (W4 = 20)
- Queue 5: 30% (W5 = 30)

Given the algorithm to get from weights to costs, the following costs are calculated:

- C0 = 32 (Smallest weight)
- C1 = 32*5/10 = 16
- C2 = 32*5/15 = 10.67 (rounded up to 11)

- C3 = 32*5/20 = 8
- C4 = 32*5/20 = 8
- C5 = 32*5/30 = 5.33 (rounded down to 5)

Due to the rounding off, these costs result in the following bandwidth distribution, which is slightly off compared to the desired distribution:

- Queue 0: 4.92%
- Queue 1: 9.85%
- Queue 2: 14.32%
- Queue 3: 19.70%
- Queue 4: 19.70%
- Queue 5: 31.51%

## 4.11.3 Shaping and DWRR Scheduling Examples

This section provides examples and additional information about the use of the egress shapers and scheduler.

**Mixing DWRR and Shaping Example**

- Port is shaped down to 500 Mbps.
- Queues 7 and 6 are strict while queue 5 through 0 are weighted.
- Queue 7 is shaped to 100 Mbps.
- Queue 6 is shaped to 50 Mbps.
- The following traffic distribution is desired for queue 5 through 0:
  Q0: 5%, Q1: 10%, Q2: 15%, Q3: 20%, Q4: 20%, Q5: 30%
- Each queue receives 125 Mbps of incoming traffic.

The following table lists the DWRR configuration and the resulting egress bandwidth for the various queues.

***Table 87 •*** **Example of Mixing DWRR and Shaping**

| Queue | Distribution of Weighted Traffic | Configuration Costs/Weights (C*n*/W*n*) | Result: Egress Bandwidth |
|---|---|---|---|
| Q0 | 5% | 32/1 | 1/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 17.2 Mbps |
| Q1 | 10% | 16/2 | 2/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 34.5 Mbps |
| Q2 | 15% | 11/2.9 | 2.9/(1+2+2.9+4+4+6.4) × (500 – Mbps – 50 Mbps) = 50.1 Mbps |
| Q3 | 20% | 8/4 | 4/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 68.9 Mbps |
| Q4 | 20% | 8/4 | 4/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 68.9 Mbps |
| Q5 | 30% | 5/6.4 | 6.4/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 110.3 Mbps |
| **Q6** | | | 50 = Mbps |
| Q7 | | | 100 = Mbps |
| **Sum:** | 100% | | **500** = **Mbps** |

**Strict and Work-Conserving Shaping Example**

- Port is shaped down to 500 Mbps.
- All queues are strict.
- All queues are shaped to 50 Mbps.
- Queues 6 and 7 are work-conserving (allowed to use excess bandwidth).
- All queues receive 125 Mbps of traffic each.

The following table lists the resulting egress bandwidth for the various queues.

***Table 88 •*** **Example of Strict and Work-Conserving Shaping**

| Queue | Result: Egress Bandwidth |
|---|---|
| Q0 | 50 Mbps |
| Q1 | 50 Mbps |
| Q2 | 50 Mbps |
| Q3 | 50 Mbps |
| Q4 | 50 Mbps |
| Q5 | 50 Mbps |
| **Q6** | 75 Mbps (Gets the last 25 Mbps of the 100 Mbps in excess not used by queue 7) |
| Q7 | 125 Mbps (Gets 75 Mbps of the 100 Mbps in excess limited only by the received rate) |
| **Sum:** | **500 Mbps** |

## 4.12 Rewriter

The switch core includes a rewriter common for all ports that determines how the egress frame is edited before transmitted. The rewriter performs the following editing:

- VLAN editing; tagging of frames and remapping of PCP and DEI.
- DSCP remarking; rewriting the DSCP value in IPv4 and IPv6 frames based on classified DSCP value.
- FCS updating.
- Precision Time Protocol timestamp updating.
- CPU extraction header insertion.

Each port module including the CPU port module has its own set of configuration in the rewriter. Each frame is handled by the rewriter one time per destination port.

### 4.12.1 VLAN Editing

The following table lists the registers associated with VLAN editing.

***Table 89 •*** **VLAN Editing Registers**

| Register | Description | Replication |
|---|---|---|
| PORT_VLAN_CFG | Port VLAN for egress port. Used for untagged set. | Per port |
| TAG_CFG | Tagging rules for port tag | Per port |
| PORT_CFG.ESO_ENA | Enable lookups in ES0. | Per port |
| PCP_DEI_QOS_MAP_CFG | Mapping table. Maps DP level and QoS class to new PCP and DEI values. | Per port per QoS per DP |

The rewriter initially pops the number of VLAN tags specified by the VLAN_POP_CNT parameter received with the frame from the classifier or VCAP IS1. Up to two VLAN tags can be popped. The rewriter itself does not influence the number of VLAN tags being popped.

For more information about each frame and destination port VCAP ES0 that is looked up using the ES0 key, see VCAP ES0, page 75. The action from an ES0 hit is used in the following to determine the frame's VLAN editing.

After popping the VLAN tags, the rewriter decides whether to push zero, one, or two new VLAN tags to the outgoing frame according to the port's tagging configuration in register TAG_CFG and the action from

a potential VCAP ES0 hit. When adding two tags, the outer tag is based on configuration in TAG_CFG while the inner tag is based on the ES0 action. When adding zero or one tag, it can either be based on TAG_CFG or ES0. Tags based on TAG_CFG settings are referred to as port tags while tags based on ES0 actions are referred to as ES0 tags.

The following table lists the possible tagging combinations:

***Table 90 •*** **Tagging Combinations**

| ES0_ACTION | TAG_CFG.TAG_CFG | Tagging action |
|---|---|---|
| No ES0 hit | 0 | No tagging. |
| No ES0 hit | 1 | Tag all frames according to the port's tagging configuration. Do not tag if VID=0 or VID=PORT_VLAN.PORT_VID. |
| No ES0 hit | 2 | Tag all frames according to the port's tagging configuration. Do not tag if VID=0. |
| No ES0 hit | 3 | Tag all frames according to the port's tagging configuration. |
| TAG_ES0=0 and TAG_TPID_SEL=0 | 0 | No tagging. |
| TAG_ES0=0 and TAG_TPID_SEL=0 | 1 | Tag all frames according to the port's tagging configuration. Do not tag if VID=0 or VID=PORT_VLAN.PORT_VID. |
| TAG_ES0=0 and TAG_TPID_SEL=0 | 2 | Tag all frames according to the port's tagging configuration. Do not tag if VID=0. |
| TAG_ES0=0 and TAG_TPID_SEL=0 | 3 | Tag all frames with port tag. |
| TAG_ES0=0 and TAG_TPID_SEL=1 | Don't care | No tagging. Overrules port settings. |
| TAG_ES0=1 | Don't care | Tag with ES0 tag only. Do not tag according to the port's tagging configuration. |
| TAG_ES0=2 | 0 | Tag with ES0 tag only. |
| TAG_ES0=2 | 1 | Tag with ES0 tag as inner tag and tag according to the port's tagging configuration as outer tag. Do not push port tag if VID=0 or VID=PORT_VLAN.PORT_VID. |
| TAG_ES0=2 | 2 | Tag with ES0 tag as inner tag and tag according to the port's tagging configuration as outer tag. Do not push port tag if VID=0. |
| TAG_ES0=2 | 3 | Tag with ES0 tag as inner tag and tag according to the port's tagging configuration as outer tag. |
| TAG_ES0=3 | Don't care | Tag with ES0 tag as inner tag and according to the port's tagging configuration as outer tag overruling tagging rule on port. |

When adding a VLAN tag, the contents of the tag header, including the TPID, is highly programmable. The starting point is the classified tag header coming from the analyzer containing a PCP, DEI, VID and tag type.

For each of the fields in the resulting tag, it is programmable how the value is determined. For the port tag, the following options are available:

**Port tag: PCP and DEI**

- Use the classified values.
- For frames generating an ES0 hit, use ES0_ACTION.PCP and ES0_ACTION.DEI; otherwise use classified values.
- Use the egress port's port VLAN (PORT_VLAN.PORT_PCP, PORT_VLAN.PORT_DEI).
- Map the DP level and QoS class to a new set of PCP and DEI using the per-port table PCP_DEI_QOS_MAP_CFG.
- Set the DEI to the DP level, independently of the preceding PCP and DEI configurations.

**Port Tag: VID**

- Use the classified VID.
- For frames generating an ES0 hit, use ES0_ACTION.VID_A_VAL; otherwise use classified VID.

**Port Tag: TPID**

- Use Ethernet type 0x8100 (C-tag)
- Use Ethernet type 0x88A8 (S-tag)
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID.
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID unless the incoming tag was a C-tag.

Similar options for the ES0 tag are available:

**ES0 tag: PCP and DEI**

- Use the classified values.
- Use ES0_action.PCP and ES0_ACTION.DEI
- Use the egress port's port VLAN (PORT_VLAN.PORT_PCP, PORT_VLAN.PORT_DEI).
- Map the DP level and QoS class to a new set of PCP and DEI using the per-port table PCP_DEI_QOS_MAP_CFG.

**ES0 tag: VID**

- Use the classified VID incremented with ES0_ACTION.VID_B_VAL.
- Use ES0_ACTION.VID_A_VAL.
- Use ES0_ACTION.VID_B_VAL.
- Use egress port's port VLAN (PORT_VLAN.PORT_VID).

**ES0 tag: TPID**

- Use Ethernet type 0x8100 (C-tag)
- Use Ethernet type 0x88A8 (S-tag)
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID.
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID unless the incoming tag was a C-tag.

## 4.12.2 DSCP Remarking

The following table lists the registers associated with DSCP remarking.

***Table 91 •*** **DSCP Remarking Registers**

| Register | Description | Replication |
|---|---|---|
| DSCP_CFG | Selects how the DSCP remarking is done | Per port |
| DSCP_REMAP_CFG | Mapping table from DSCP to DSCP for DP level = 0. | None |
| DSCP_REMAP_DP1_CFG | Mapping table from DSCP to DSCP for DP level = 1. | None |

The rewriter can remark the DSCP value in IPv4 and IPv6 frames, that is, write a new DSCP value to the DSCP field in the frame.

If a port is enabled for DSCP remarking (DSCP_CFG.DSCP_REWR_CFG), the new DSCP value is derived by using the classified DSCP value from the analyzer (the basic classification or the VCAP IS1)

in the ingress port. This DSCP value can be mapped before replacing the existing value in the frame. The following options are available:

- No DSCP remarking - Leave the DSCP value in the frame untouched.
- Update the DSCP value in the frame with the value received from the analyzer
- Update the DSCP value in the frame with the value received from the analyzer remapped through DSCP_REMAP_CFG. This is done independently of the value of the drop precedence level.
- Update the DSCP value in the frame with the value received from the analyzer remapped through DSCP_REMAP_CFG or DSCP_REMAP_DP1_CFG dependent on the drop precedence level. This enables one mapping for green frames and another for yellow frames so that the resulting DSCP value can reflect the color of the frame.

Additionally, the IP checksum is updated for IPv4 frames. Note that the IPv6 header does not contain a checksum. As a result, checksum updating does not apply for IPv6 frames.

DSCP remarking is not possible for frames where PTP timestamps are also generated and is automatically disabled.

### 4.12.3 FCS Updating

The following table lists the registers associated with FCS updating.

***Table 92 •*** **FCS Updating Registers**

| Register | Description | Replication |
|---|---|---|
| PORT_CFG.FCS_UPDATE_NONCPU_CFG | FCS update configuration for non-CPU injected frames. | Per port |
| PORT_CFG.FCS_UPDATE_CPU_ENA | FCS update configuration for CPU injected frames. | Per port |

The rewriter updates a frame's FCS when required or instructed to do so. Different handling is available for frames injected by the CPU and for all other frames.

For non-CPU injected frames, the following update options are available:

- Never update the FCS.
- Conditional update - Update the FCS if the frame was modified due to PTP timestamping, VLAN tagging or DSCP remarking.
- Always update the FCS.

Additionally, the rewriter can update the FCS for all frames injected from the CPU through the CPU injection queues in the CPU port module:

- Never update the FCS.
- Always update the FCS.

### 4.12.4 CPU Extraction Header Insertion

The following table lists the registers associated with CPU extraction header insertion.

***Table 93 •*** **CPU Extraction Header Insertion Registers**

| Register | Description | Replication |
|---|---|---|
| PORT_CFG.IFH_INSERT_ENA | Enables insertion of the CPU extraction header. | Per port |
| PORT_CFG.IFH_INSERT_MODE | Configures the position of the CPU extraction header. | Per port |

Any port in the switch core can request the rewriter to insert a CPU extraction header in the frame before transmission. For more information about the contents of the CPU extraction header, see CPU Extraction and Injection, page 235.

The CPU extraction header can be placed before the DMAC or right after the SMAC. When inserting the header, the frame is extended with eight bytes. Note that the FCS is only updated when the header is inserted after the SMAC.

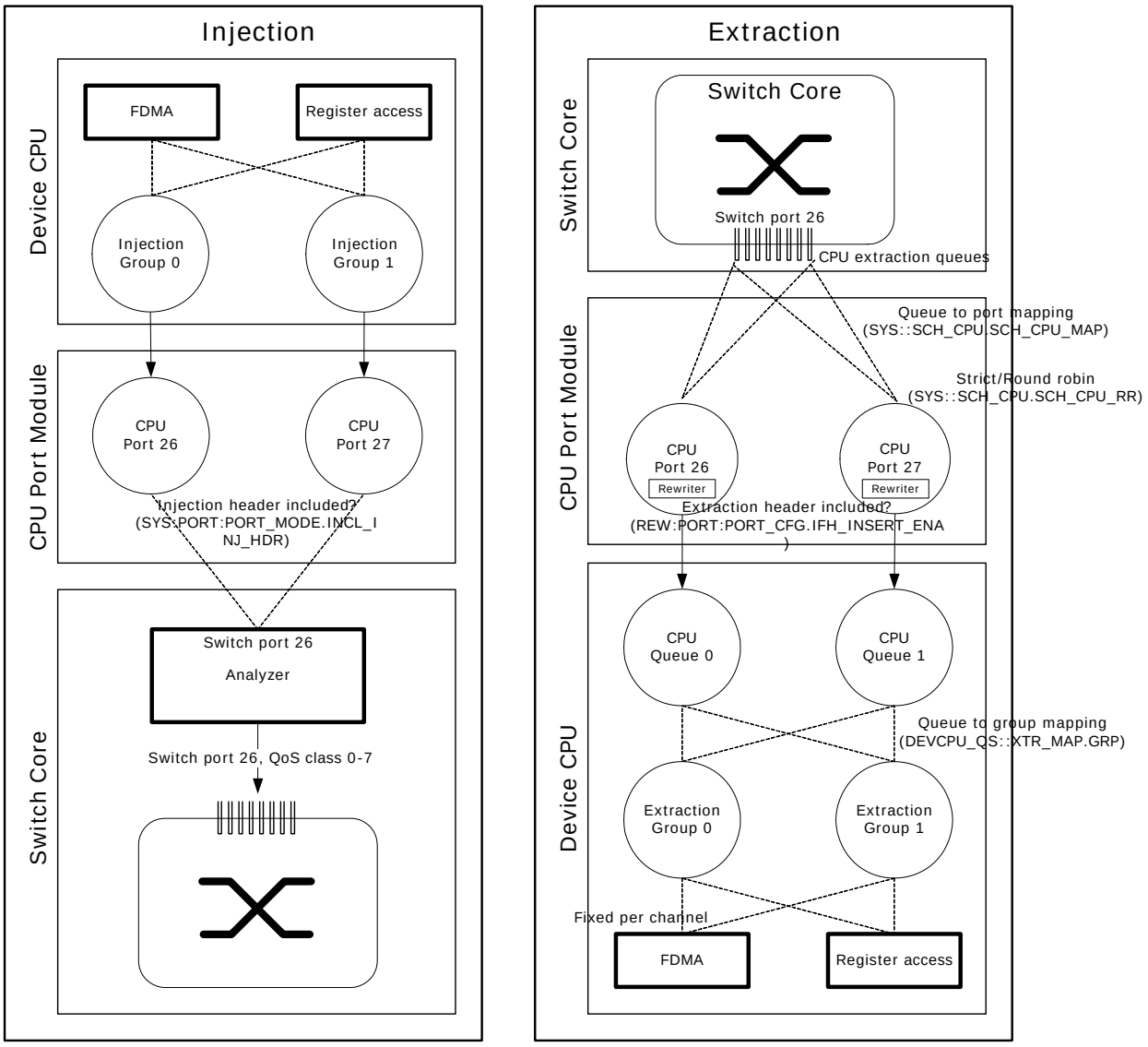
The insertion of the CPU extraction header is the last editing in the rewriter. This implies that any VLAN tags in the frame will appear after the extraction header.

## 4.13 CPU Port Module

The CPU port module connects the switch core to the CPU system so that frames can be injected from or extracted to the CPU. It is also possible to use a regular front port as a CPU port. This is known as a Network Processor Interface (NPI).

The following illustration shows how the switch core interfaces to the CPU system through the CPU port module for injection and extraction of frames.

***Figure 35 •*** **CPU Injection And Extraction**



### 4.13.1 Frame Extraction

The following table lists the registers associated with frame extraction.

***Table 94 •*** **Frame Extraction Registers**

| Register | Description | Replication |
|---|---|---|
| SYS::SCH_CPU.SCH_CPU_MAP | Configuration of mapping of extraction queues to CPU ports | Per CPU port (ports 26 and 27) |
| SYS::SCH_CPU.SCH_CPU_RR | Configuration of CPU scheduler | Per CPU port (ports 26 and 27) |
| REW:PORT:PORT_CFG.IFG_INSERT_ENA | Enables insertion of extraction header | Per CPU port (port 26 and 27) |

In the switch core, extracted frames are forwarded to one of the eight CPU extraction queues. Each of these queues is mapped to one of two CPU ports (port 26 and port 27) through SYS::SCH_CPU.SCH_CPU_MAP. For each CPU port, there is a scheduler working either in strict mode or round robin, which selects between the CPU extraction queues mapped to the same CPU port (SYS::SCH_CPU.SCH_CPU_RR). In strict mode, higher queue numbers are preferred over smaller queue numbers. In round robin, all queue are serviced one after another.

The two CPU ports contain the same rewriter as regular front ports. The rewriter modifies the frames before sending them to the CPU. In particular, the rewriter inserts an extraction header (REW::PORT:PORT_CFG.IFH_INSERT_ENA), which contains relevant side band information about the frame such as the frame's classification result (VLAN tag information, DSCP, QoS class) and the reason for sending the frame to the CPU. For more information about the rewriter, see Rewriter, page 116.

The device CPU contains the functionality for reading out the frames. This can be done through the frame DMA or regular register access.

The following table lists the contents of the CPU extraction header.

***Table 95 •*** **CPU Extraction Header**

| Field | Bit | Width | Description |
|---|---|---|---|
| SIGNATURE | 56 | 8 | Must be 0xFF. |
| SRC_PORT | 51 | 5 | The port number where the frame was received (0-26). |
| DSCP | 45 | 6 | The frame's classified DSCP value.<br>If the frame is hardware timestamped (frame has hit a rule in IS2 with PTP_ENA), the DSCP field contains the timestamp identifier provided by the analyzer, see **Two-Step Timestamping**, page 128. |
| ACL_IDX | 37 | 8 | If ACL_HIT is set, this value is the entry number of the rule hit in IS2. If both IS2 lookups hit a rule which copy the frame to the CPU, the second lookup's entry number is used. |
| SFLOW_ID | 32 | 5 | sFlow sampling ID.<br>0-26: Frame was SFlow sampled by a Tx sampler on port given by SFLOW_ID.<br>27: Frame was SFlow sampled by an RX sampler on port given by SRC_PORT.<br>28-30: Reserved.<br>31: Frame was not SFlow sampled. |
| ACL_HIT | 31 | 1 | Set if frame has hit a rule in IS2, which copies the frame to the CPU (IS2 actions CPU_COPY_ENA or HIT_ME_ONCE). ACL_IDX contains the IS2 entry number. |

*Table 95 •* **CPU Extraction Header *(continued)***

| Field | Bit | Width | Description |
|---|---|---|---|
| DP | 30 | 1 | The frame's drop precedence (DP) level after policing. |
| LRN_FLAGS | 28 | 2 | The source MAC address learning action triggered by the frame.<br>0: No learning.<br>1: Learning of a new entry.<br>2: Updating of an already learned unlocked entry.<br>3: Updating of an already learned locked entry. |
| CPU_QUEUE | 20 | 8 | CPU extraction queue mask (one bit per CPU extraction queue). Each bit set implies the frame was subjected to CPU forwarding to the specific queue. |
| QOS_CLASS | 17 | 3 | The frame's classified QoS class. |
| TAG_TYPE | 16 | 1 | The tag information's associated Tag Protocol Identifier (TPID). The definitions are:<br>0: C-tag: EtherType = 0x8100.<br>1: S-tag: EtherType = 0x88A8 or custom value. |
| PCP | 13 | 3 | The frame's classified PCP. |
| DEI | 12 | 1 | The frame's classified DEI. |
| VID | 0 | 12 | The frame's classified VID. |

## 4.13.2 Frame Injection

The following table lists the registers associated with frame injection.

*Table 96 •* **Frame Injection Registers**

| Register | Description | Replication |
|---|---|---|
| SYS:PORT:PORT_MODE.INCL_INJ_HDR | Enable parsing of injection header | Per CPU port (ports 26 and 27) |
| SYS:PORT:EQ_PREFER_SRC | Enable preferred arbitration of the CPU port (port 26) over front ports | CPU port (port 26 only) |

The CPU injects frames through the two CPU injection groups independent of each other. The injection groups connect to the two CPU ports (port 26 and port 27) in the CPU port module. In CPU port module, each of the two CPU ports have dedicated access to the switch core. Inside the switch core, all CPU injected frames are seen as coming from CPU port (port 26). This implies that both CPU injection groups consume memory resources from the shared queue system for port 26 and that analyzer configuration for port 26 are applied to all frames.

In the switch core, the CPU port can be preferred over other ingress ports when transferring frames to egress queues by enabling precedence of the CPU port (SYS::EQ_PREFER_SRC).

The first eight bytes of a frame written to a CPU injection group is an injection header containing relevant side band information about how the frame must be processed by the switch core. The CPU ports must be enabled to expect the CPU injection header (SYS:PORT:INCL_INJ_HDR).

On a per-frame basis, the CPU controls whether frames injected through the CPU port module are processed by the analyzer. If the frame is processed by the analyzer, it is sent through the processing steps to calculate the destination ports for the frame. If analyzer processing is not selected, the CPU can specify the destination port set and related information to fully control the forwarding of the frame. For more information about the analyzer's processing steps, see Forwarding Engine, page 92.

The contents of the CPU injection header is listed in the following table.

***Table 97 •*** **CPU Injection Header**

| Field | Bit | Width | Description |
|---|---|---|---|
| BYPASS | 63 | 1 | When this bit is set, the analyzer processing is skipped for this frame. The destination set is specified in DEST and CPU_QUEUE. Forwarding uses the QOS_CLASS, and the rewriter uses the tag information (POP_CNT, TAG_TYPE, PCP, DEI, VID) for rewriting actions.<br>When this bit is cleared, the analyzer determines the destination set, QoS class, and VLAN classification for the frame through normal frame processing including lookups in the MAC table and VLAN table. |
| PTP | 61 | 2 | The frame's Precision Time Protocol action. The definitions are:<br>0: No PTP action.<br>1: One-step; update the residence time in the PTP protocol.<br>2: Two-step; register the residence time in the PTP timestamp queue using the PTP_ID as identifier.<br>3: Both one-step and two-step. Used when BYPASS = 1. |
| PTP_ID | 59 | 2 | The PTP identifier used for two-step PTP actions. The CPU can only use from IDs 0 through 3. Used when BYPASS = 1. |
| DEST | 32 | 27 | This is the destination set for the frame. DEST[26] is the CPU. Used when BYPASS = 1. |
| RESERVED | 30 | 2 | Unused. |
| POP_CNT | 28 | 2 | Number of VLAN tags that must be popped in the rewriter before adding new tags. Used when BYPASS = 1.<br>0: No tags must be popped.<br>1: One tag must be popped.<br>2: Two tags must be popped.<br>3: Disable rewriting of VLAN tags and DSCP value. The FCS is still updated. |
| CPU_QUEUE | 20 | 8 | CPU extraction queue mask (one bit per CPU extraction queue). Each bit set implies the frame must be forwarded by the CPU to the specific queue.<br>Used when BYPASS = 1 and DEST[26] = 1. |
| QOS_CLASS | 17 | 3 | The frame's classified QoS class.<br>Used when BYPASS = 1. |
| TAG_TYPE | 16 | 1 | The tag information's associated Tag Protocol Identifier (TPID). Used when BYPASS = 1.<br>0: C-tag: EtherType = 0x8100.<br>1: S-tag: EtherType = 0x88A8 or custom value. |
| PCP | 13 | 3 | The frame's classified PCP. Used when BYPASS = 1. |
| DEI | 12 | 1 | The frame's classified DEI. Used when BYPASS = 1. |
| VID | 0 | 12 | The frame's classified VID. Used when BYPASS = 1. |

### 4.13.3 Network Processor Interface (NPI)

The following table lists the registers associated with the network processor interface.

***Table 98 •*** **Network Processor Interface Registers**

| Register | Description | Replication |
|---|---|---|
| SYS::EXT_CPU_CFG | Configuration of the NPI port number and configuration of which CPU extraction queues are redirected to the NPI. | None |
| REW:PORT:PORT_CFG.IFG_INSERT_ENA | Enables insertion of extraction header | Per port |
| SYS:PORT:PORT_MODE.INCL_INJ_HDR | Configuration of NPI ingress mode. | Per port |

Any front port can be configured as a network processor interface through which frames can be injected from and extracted to an external CPU. Only one port can be an NPI at the same time. SYS::EXT_CPU_CFG.EXT_CPU_PORT holds the port number of the NPI.

A dual CPU system is possible where both the internal and the external CPU are active at the same time. Through SYS::EXT_CPU_CFG.EXT_CPUQ_MSK, it is configurable to which of the eight CPU extraction queues are directed to the internal CPU and which are directed to external CPU. A frame can be extracted to both the internal CPU and the external CPU if the frame is extracted for multiple reasons.

A frames being extracted to the external CPU can have the CPU extraction header inserted in front of the frame (REW:PORT:PORT_CFG.IFG_INSERT_ENA), and a frame being injected to the switch core can have the CPU injection header inserted in front of the frame (SYS:PORT:PORT_MODE.INCL_INJ_HDR).

Through the BYPASS field in the CPU injection header, the external CPU can control forwarding of injected frames by either letting the frame analyze and forward accordingly or directly specifying the destination set

## 4.14 Hardware Timestamping for AVB

Hardware timestamping provides nanosecond-accurate frame arrival and departure time stamps, which are used to obtain high precision timing synchronization and timing distribution, as well as significantly better accuracy in performance monitoring measurements than what is obtained from pure software implementations.

All frames are Rx timestamped on arrival with a 32-bit timestamp value using a hardware timer (timestamper) implemented in the Media Access Control (MAC) block. The Rx timestamper provides high timestamp accuracy relative to actual arrival time of the first byte of the frame from the PHY device. Within the VCAP IS2, it is decided if the frame and associated Rx timestamp must be redirected or copied to CPU for processing. The frame is forwarded as normal otherwise.

The VCAP IS2 also decides if a Tx timestamp must be triggered for a frame. Given the Rx and Tx timestamps, the frame's residence time inside the switch is calculated. The residence time can be stored in a timestamp queue for the CPU to access (two-step timestamping) or the residence time can be used to update the residence time field inside Precision Time Protocol frames (one-step timestamping).

The Tx timestamper is located at the transmit side of the MAC block as close to the PHY device as possible and provides high accuracy of timestamp relative to when the first byte of the frame is actually transmitted to the PHY.

The devices also implement a time of day counter with microsecond accuracy. The time of day counter is derived from a one-second timer. The one-second timer generates a pulse per second and is derived from an adjusted system clock.

### 4.14.1 Timestamp Classification

Frames requiring Rx or Tx timestamping are identified by VCAP IS2. The IS2 action that triggers timestamping is PTP_ENA, where PTP_ENA[0] enables one-step timestamping, and PTP_ENA[1] enables two-step timestamping.

IS2 can be configured to identify the following frame formats from IEEE 1588-2008:

- Transport of PTP over User Datagram Protocol over Internet Protocol Version 4
- Transport of PTP over User Datagram Protocol over Internet Protocol Version 6
- Transport of PTP over IEEE 802.3/Ethernet

### 4.14.2 One-Second Timer

The one-second timer generates one synchronization pulse per second, which is used for the time of day counter. The one-second timer and the time of day counter are located in the CPU System block.

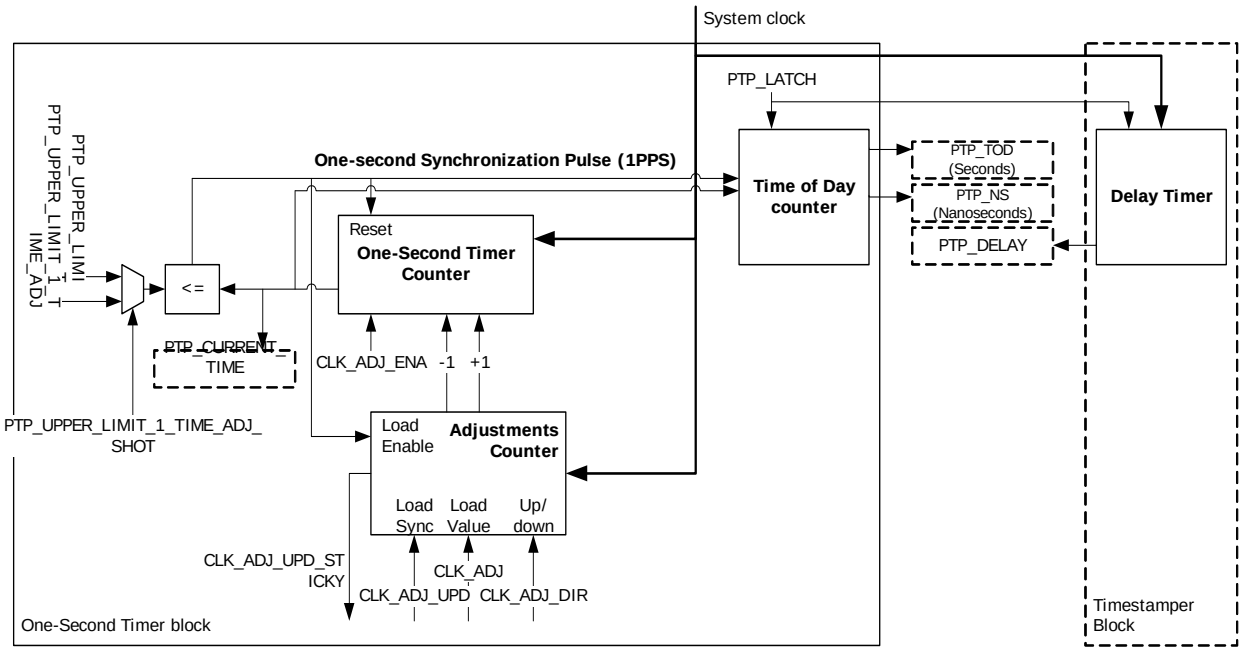
The registers listed in the following table control and monitor the one-second timer.

***Table 99 •*** **One-Second Timer Registers**

| Target:Register_group:Register.field | Description | Replication |
|---|---|---|
| DEVCPU_GCB::PTP_MISC_CFG | GPIO configuration of hardware timer | 1 |
| DEVCPU_GCB::PTP_UPPER_LIMIT_CFG | One-second counter configuration | 1 |
| DEVCPU_GCB::PTP_UPPER_LIMIT_1_TIME_ADJ_CFG | One-second counter configuration | 1 |
| DEVCPU_GCB::CLK_ADJ_CFG | One-second Counter adjustment configuration | 1 |
| DEVCPU_GCB::PTP_SYNC_INTR_ENA_CFG | Interrupts control | 1 |
| DEVCPU_GCB::PTP_CURRENT_TIME_STAT | One-second counter statistics. Current count value. | 1 |
| DEVCPU_GCB::EXT_SYNC_CURRENT_TIME_STAT | One-second counter statistics.. | 1 |
| DEVCPU_GCB::PTP_EVT_STAT | One-second timer event statistics. | 1 |

The one-second timer block diagram is shown in the following illustration.

***Figure 36 •*** **One-Second Timer Block Diagram**




DEVCPU_GCB::PTP_MISC_CFG.PTP_ENA enables the one-second timer and must be set for one-second timer synchronization pulse generation.

By default, the one-second timer synchronization pulse is generated internally and with a frequency of one pulse per second (1 PPS) derived from the system clock. Other one-second timer synchronization pulse frequencies are obtained using register DEVCPU_GCB::PTP_UPPER_LIMIT_CFG. Every time a one-second timer synchronization pulse is generated, a sticky bit is set (DEVCPU_GCB::PTP_EVT_STAT.SYNC_STAT) and an interrupt is generated if DEVCPU_GCB::PTP_SYNC_INTR_ENA_CFG.SYNC_STAT_ENA is enabled.

#### 4.14.2.1 One-Second Timer Counter Adjustments

If a one time correction to the one-second timer synchronization pulse is required, the correction time value must be written into register DEVCPU_GCB::PTP_UPPER_LIMIT_1_TIME_ADJ_CFG.PTP_UPPER_LIMIT_1_TIME_ADJ and one shot is enabled in register DEVCPU_GCB::PTP_UPPER_LIMIT_1_TIME_ADJ_CFG.PTP_UPPER_LIMIT_1_TIME_ADJ_SHOT.

The one-second timer can also be controlled by issuing counter corrections to the one-second timer counter. One-second timer counter corrections are enabled in register DEVCPU_GCB::CLK_ADJ_CFG.CLK_ADJ_ENA.

Corrections to the one-second timer counter is controlled by the adjustments counter. The adjustments counter issues ±1 corrections to the one-second timer counter. The time period between one-second timer corrections is determined by the load value of the adjustments counter. Time periods between corrections ranges from nanoseconds to one second.

The adjustments counter operates as follows:

- When the counter value of the adjustments counter equals the load value (DEVCPU_GCB::CLK_ADJ_CFG.CLK_ADJ), a one tick correction is generated.
- Up or down corrections are determined by DEVCPU_GCB::CLK_ADJ_CFG.CLK_ADJ_DIR.
- The DEVCPU_GCB::CLK_ADJ_CFG.CLK_ADJ_UPD register controls whether a load value change takes immediate effect or whether it is synchronized to the next one-second timer synchronization pulse.
- When the load value change occurs, a sticky bit is set (DEVCPU_GCB::PTP_EVT_STAT.CLK_ADJ_UPD_STICKY). This sticky does not gate future

updates to the load value and is informative only. The adjustment counter is reset by loading all zeros.

## 4.14.3 Delay Timer

This section explains the functions of the hardware timestamping module. The following table lists the registers associated with the delay timer.

***Table 100 •*** **Hardware Timestamping Registers**

| Register | Description | Replication |
|---|---|---|
| SYS:PORT:PTP_CFG | Enabling of Tx handling. Rx and Tx timestamp adjustments. | Per port |
| SYS:PORT:PTP_DELAY | Timestamp value in timestamp queue | Per port |
| SYS:PORT:PTP_NXT | Advancing the timestamp queue | Per port |
| SYS:PORT:PTP_STATUS | Timestamp queue status and entry data | Per port |
| ANA::PTP_ID_HIGH | Release of timestamp identifiers, values 32 through 63. | None |
| ANA::PTP_ID_LOW | Release of timestamp identifiers, values 0 through 31. | None |

Each port module contains a hardware timestamping module that measures arrival and departure times based on a free-running delay timer. The delay timer is derived from the system clock and is independent of the one-second timer. The two timing domains can be correlated using the time of day latching. For more information, see Time of Day Counter, page 129.

### 4.14.3.1 Rx And Tx Timestamps

When the MAC block determines that a new frame has arrived, the Rx timestamper generates a timestamp, which follows the frame all the way to the Tx side. At the Tx side, the Tx timestamper generates a timestamp only if the frame has matched a VCAP IS2 entry with a PTP_ENA action set.

The arrival and departure times can be shifted in time so that the timestamps match the exact arrival and departure times of the first byte in the frame (SYS:PORT:PTP_CFG.IO_RX_DELAY, SYS:PORT:PTP_CFG.IO_TX_DELAY). Rx and Tx can be adjusted individually. The resulting arrival and departure times are given as:

- Arrival time — Sampling of delay timer minus SYS:PORT:PTP_CFG.IO_RX_DELAY
- Departure time — Sampling of delay timer plus SYS:PORT:PTP_CFG.IO_TX_DELAY

When Tx timestamping is performed, the frame's residence time is calculated as departure time minus arrival time. The residence time can be handled in two different ways based on the action received from the IS2.

#### 4.14.3.2 One-Step Timestamping

If the IS2_ACTION.PTP_ENA[0] action is set, one-step timestamping is performed. This only applies to the following frame formats:

- PTP frames over UDP over IPv4 with zero, one, or two VLAN tags
- PTP frames over UDP over IPv6 with zero, one, or two VLAN tags
- PTP frames over IEEE 802.3/Ethernet with zero, one, or two VLAN tags

The number of VLAN tags here is defined as the number of VLAN tags after the rewriter has completed the VLAN editing of the frame in terms of popping and pushing VLAN tags.

When performing one-step timestamping, the residence time is added to the frame's PTP correction field by:

1. Reading the correction field in the received PTP header
2. Adding the frame's residence time
3. Writing the result back into the frame's correction field.

When changing the correction field in PTP frames over UDP, the UDP checksum is simultaneously cleared (set to zero). This is the case for both IPv4 and IPv6 frames.

One-step timestamping can be disabled per egress port using SYS:PORT:PTP_CFG.PTP_1STEP_DIS. This setting overrules the IS2 action.

#### 4.14.3.3 Two-Step Timestamping

Two-step timestamping is performed if the IS2_ACTION.PTP_ENA[1] action is set. This action applies to any frame, because the frame itself is not modified. The residence time is stored in a timestamp FIFO queue, which the CPU can access (SYS:PORT:PTP_STATUS). The timestamp is common for all egress ports and can contain up to 128 timestamps. Each entry in the timestamp queue contains the following fields:

- SYS:PORT:PTP_STATUS.PTP_MESS_VLD: A 1-bit valid bit meaning the entry is ready for reading.
- SYS:PORT:PTP_STATUS.PTP_MESS_ID: A 6-bit timestamp identifier. A unique timestamp identifier is assigned to each frame for which one or more Tx timestamps are generated. The timestamp identifier is also available in the CPU extraction header for frames extracted to the CPU. The timestamp identifier overloads the DSCP value in the CPU extraction header. For more information about the CPU extraction header, see Table 95, page 121. By providing the timestamp identifier in both the timestamp queue and in the extracted frames, the CPU can correlate which timestamps belong to which frames. Note that timestamp identifier value 63 implies that no free identifier could be assigned to the frame. The timestamp entry can therefore not be trusted.
- SYS:PORT:PTP_STATUS.PTP_MESS_TXPORT: The port number where the frame is transmitted. When transmitting a frame on multiple ports, there are generated multiple entries in the timestamp queue. Each entry uses the same timestamp identifier but with different Tx port numbers.
- SYS:PORT:PTP_DELAY: The frame's residence time when the Tx port is a front port or the frame's arrival time when the Tx port is the CPU port.

The timestamp queue is a simple FIFO that can be read by the CPU. The timestamp queue provides the following handles for reading:

- Overflow of the queue is signaled through SYS:PORT:PTP_STATUS.PTP_OVFL. Overflow implies that one or more timestamps could not be enqueued due to all 128 entries being in use. Timestamp not enqueued are lost.
- The head-of-line entry is read through SYS:PORT:PTP_STATUS and SYS:PORT:PTP_DELAY.
- Writing to the one-shot register SYS:PORT_PTP_NXT removes the current head-of-line entry and advances the pointer to the next entry in the timestamp queue.

When two-step Tx timestamping is performed for a frame destined for the CPU extraction queues, the frame's arrival timestamp is enqueued in the timestamp queue instead of the frame's residence time. This enables the CPU to acknowledge the arrival time of the frame and simultaneously sample the delay timer when the frame is extracted from the CPU extraction queues to calculate the exact residence time from the frame enters the switch to the CPU receives the frame.

The timestamp identifiers can take values between 0 to 63. Value 63 implies that all values 0-62 are in use. Values 0 – 3 are pre-assigned to the CPU to be used for injection of frames. The remaining values are assigned by the analyzer to frames requesting timestamping through the VCAP IS2 action. The assigned values must be released again by the CPU by writing to the corresponding bit in ANA::PTP_ID_HIGH (values 32 through 63) or ANA::PTP_ID_LOW (values 0 through 31). The CPU releases a timestamp identifier when it has read the anticipated timestamp entries from the timestamp queue. Note that multicasted frames generate a timestamp entry per egress port using the same timestamp identifier. Each of these entries must be read before the timestamp identifier is released.

Two-step timestamping can be disabled per egress port using SYS:PORT:PTP_CFG.PTP_2STEP_DIS. This setting overrules the IS2 action.

#### 4.14.3.4 DSCP Remarking

If a frame is being timestamped, DSCP remarking is automatically disabled for the frame.

### 4.14.4 Time of Day Counter

The time of day counter holds a 32 bits seconds counter and a 28 bits nanoseconds counter. The nanoseconds counter is derived from the one-second timer counter, and the seconds counter increments based on the one-second synchronization pulse.

The registers listed in the following table are used for controlling and monitoring the time of day counter.

***Table 101 •*** **Time of Day Counter Registers**

| Target:Register_group:Register.field | Description | Replication |
|---|---|---|
| SYS::PTP_TOD_SECS | Latched value of time of day counter (seconds) | None |
| SYS::PTP_TOD_NANOSECS | Latched value of time of day counter (nanoseconds) | None |
| SYS::PTP_DELAY | Latched value of delay timer | None |
| SYS::PTP_TIMER_CTRL | Control of latching | None |

The time of day counter is enabled through SYS::PTP_TIMER_CTRL.PTP_TIMER_ENA. The 32-bit seconds counter can be reset (SYS::PTP_TIMER_CTRL.PTP_TOD_RST), and the 28-bit nanoseconds counter directly follows the one-second timer counter.

The time of day counter and the delay timer used in the port modules for timestamping can be latched at the same time so that the timestamps in frames can be correlated to day using the one-shot SYS::PTP_TIMER_CTRL.PTP_LATCH. The results of the latching are stored in the following registers and contain counter values from the same point in time:

- Delay timer: SYS::PTP_DELAY
- Time of day counter (seconds): SYS::PTP_TOD_SECS
- Time of day counter (nanoseconds): SYS::PTP_TOD_NANOSECS

## 4.15 Clocking and Reset

The following table lists the registers associated with clocking and reset.

***Table 102 •*** **Clocking and Reset Registers**

| Target:Register_group:Register.field | Description | Replication |
|---|---|---|
| HSIO::PLL5G_CFG0 | LCPLL configuration | None |
| HSIO::PLL5G_STATUS0 | LCPLL status | None |
| DEVCPU_GCB:: SOFT_CHIP_RST | Reset of the internal copper PHYs or the entire device | None |
| DEVCPU_GCB::SOFT_DEVCPU_RST | Reset of the extraction and injection modules | None |
| CFG::RESET | CPU reset configuration | None |

The LCPLL provides the clocks used by the SerDes, the central part of the switch core, and the VCore-III CPU system.

The reference clock for the LCPLL (REFCLK_P and REFCLK_N pins) is either differential or single-ended. The frequency can be 25 MHz, 125 MHz, or 156.25 MHz. For more information about the reference clock frequency selections, see the Pins by Function section for the appropriate device.

For more information about reference clock options, see Reference Clock, page 733.

A global software reset is performed with DEVCPU_GCB::SOFT_CHIP_RST.

For more information about the configuration of the CPU frequency and software reset options when using the V-Core-III, see Clocking and Reset, page 133.

For more information about the clock and reset configuration for the Ethernet interfaces in the port modules, see MAC, page 16, SERDES1G, page 22, and SERDES6G, page 26. The MAC clock domains are not included in the global reset.

# 5 VCore-III System and CPU Interface

This section provides information about the functional aspects of blocks and the interfaces related to the VCore-III on-chip microprocessor system.

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices contain a powerful VCore-III CPU system that is based on an embedded MIPS24KEc-compatible microprocessor and a high bandwidth DMA engine. The VCore-III system can control the devices independently or it can support an external CPU, relieving the external CPU of the otherwise time-consuming tasks of transferring frames, maintaining the switch core, and handling networking protocols.

When the VCore-III CPU is enabled, it either boots up independently from Flash or a code-image can be manually loaded and started from an external CPU.

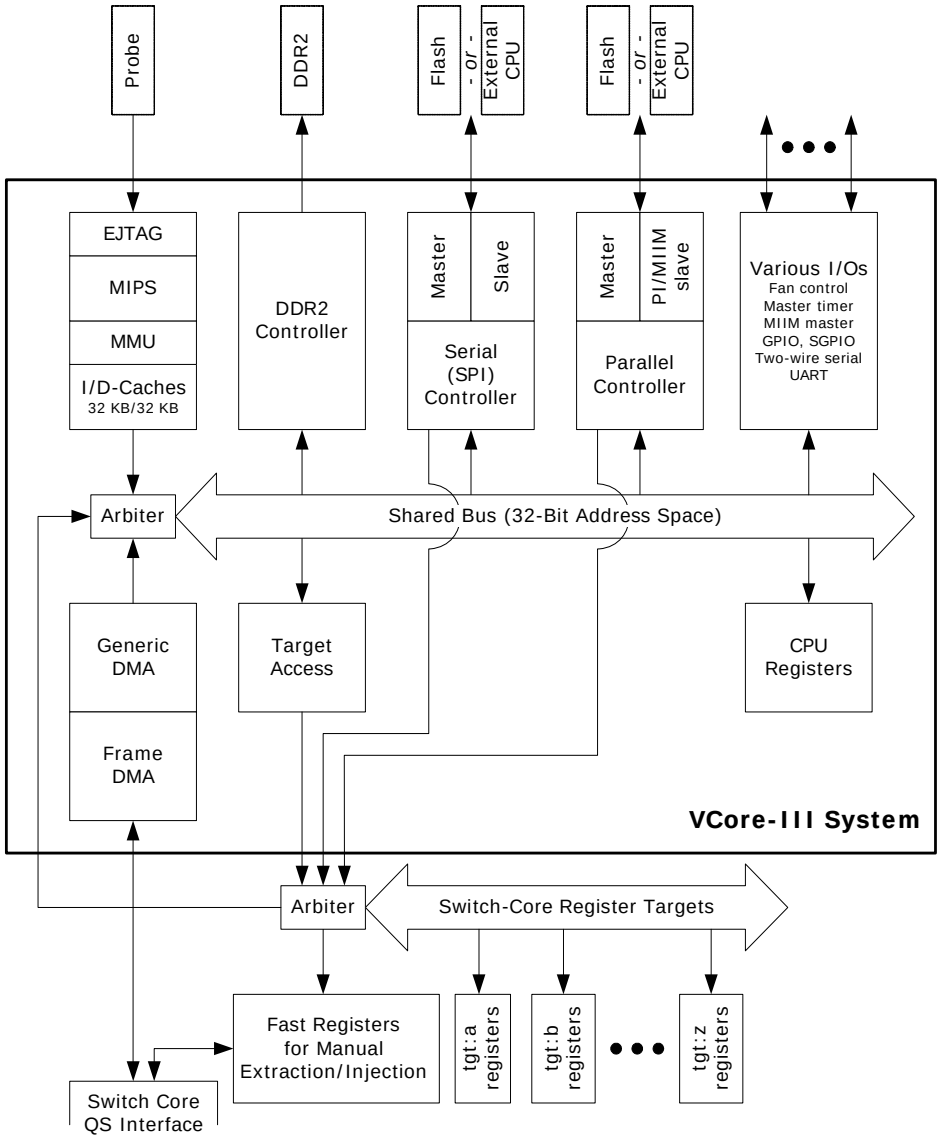
An external CPU can be connected to the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices through the serial interface (SI), parallel interface (PI), or dedicated MIIM slave interface. When the VCore-III CPU is enabled and boots up from Flash, the SI is reserved as boot interface and cannot be used by an external CPU.

The VCore-III CPU and the external CPUs can access internal chip registers for configuration, monitoring, and collecting statistics.

The VCore-III system includes a number of functional blocks and registers that are tightly coupled to the VCore-III CPU. The external CPU can access these blocks and register through an indirect addressing scheme. The registers are available when the VCore-III CPU is enabled or disabled.

The following illustration shows how the serial, parallel, and MIIM controllers operate in either master or slave mode. When the VCore-III CPU is enabled, it forces the boot interface to master mode. An interface in slave mode allows an external CPU access to register targets inside the device.

*Figure 37 •* **VCore-III System Block Diagram**




## 5.1 VCore-III Configurations

The following table summarizes possible VCore-III configurations.

*Table 103 •* **VCore-III Configurations**

| **Level of Strapping Pins** | | | |
|---|---|---|---|
| **VCore_CFG[2]** | **VCore_CFG[1]** | **VCore_CFG[0]** | **Behavior** |
| Endian mode | 0 | 0 | MIPS is enabled and boots up from SI. |
| Endian mode | 0 | 1 | Automatic boot is disabled by forcing the MIPS into reset.<br>SI slave mode is enabled.<br>The MIPS can be manually started from the DDR. |

*Table 103 •* **VCore-III Configurations *(continued)***

| Level of Strapping Pins | | | |
|---|---|---|---|
| **VCore_CFG[2]** | **VCore_CFG[1]** | **VCore_CFG[0]** | **Behavior** |
| Endian mode | 1 | 0 | Automatic boot is disabled by forcing the MIPS into reset.<br>PI and SI slave modes are enabled.<br>The MIPS can be manually started from the DDR or Flash on the SI interface. |
| Endian mode | 1 | 1 | Automatic boot is disabled by forcing the MIPS into reset.<br>MIIM and SI slave modes are enabled.<br>The MIPS can be manually started from the DDR or Flash on the SI interface. |

The VCORE_CFG pins control the behavior of the VCore-III system. The VCore-III CPU operates either in little endian or big endian mode. To enable big endian mode, tie the VCORE_CFG[2] configuration input high. In big endian mode, register access must be byte-swapped when reading and writing. For more information, see the API documentation on www.vitesse.com.

The EJTAG interface of the VCore-III CPU and the Boundary Scan JTAG controller are both multiplexed onto the JTAG interface of the device. When the VCore_ICE_nEn pin is low, the MIPS's EJTAG controller is selected. When the VCore_ICE_nEn pin is high, the Boundary Scan JTAG controller is selected.

## 5.2 Clocking and Reset

The following table lists the registers associated with clocking and reset.

*Table 104 •* **Clocking and Reset Configuration Registers**

| Register | Description |
|---|---|
| PLL5G_CFG0 | Configures VCore-III CPU frequency |
| RESET | VCore-III reset configuration and release of specific blocks from reset |
| SOFT_CHIP_RST | Resets configuration |
| WDT | Watchdog timer configuration and status |

The frequency of the VCore-III CPU is controlled by PLL5G_CFG0.CPU_CLK_DIV. The VCore-III system operates at the same frequency as the VCore-III CPU. The frequency can be changed on-the-fly while the VCore-III CPU is running. When using devices that require a constant clock frequency during normal operation (for example, UART), it is recommended that software configure the clock frequency once during boot up.

The frequency of the VCore-III CPU must not exceed the speed of the available DDR2 SDRAMs. The DDR frequency is locked to half the VCore-III CPU frequency. For example, if DDR400 is used (with a maximum clock of 200 MHz), the maximum VCore-III CPU frequency, when equipped with DDR400 SDRAM, is 312.5 MHz.

The VCore-III CPU (including the VCore-III system) can be soft-reset by setting RESET.CORE_RST_FORCE. By default, this resets both the VCore-III CPU and the VCore-III system. The VCore-III system can be excluded from a soft reset by setting RESET.CORE_RST_CPU_ONLY;soft-reset using CORE_RST_FORCE only then resets the VCore-III CPU. The Frame DMA must be disabled prior to a soft reset of the VCore-III system. When CORE_RST_CPU_ONLY is set, the Frame DMA and memory system are unaffected by a soft reset and continue to operate throughout soft reset of the VCore-III CPU.

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices can be soft-reset by using SOFT_CHIP_RST.SOFT_CHIP_RST, which by default, resets the entire device. The VCore-III system

and CPU can be protected from a chip-level soft reset by configuring RESET.CORE_RST_PROTECT. In this case, a chip-level soft reset is applied to all other blocks except the VCore-III system and CPU. When protecting the VCore-III system and CPU from a soft reset, the Frame DMA must be disabled prior to a chip-level soft reset.

The GPIO alternate modes are reset to the default values when performing chip-level soft reset. This must be taken into account when the VCore-III system is protected from chip-level soft reset (by means of RESET.CORE_RST_PROTECT).

When automatic booting of the VCore-III CPU is disabled using the VCORE_CFG pins, the VCore-III CPU can be manually released through RESET.CPU_RELEASE.

### 5.2.1 Watchdog Timer

The VCore-III system has a built-in watchdog timer (WDT) with a time-out cycle of two seconds. The watchdog timer is enabled, disabled, or reset through the WDT register. The watchdog timer is disabled by default.

After the watchdog timer is enabled, it must be regularly reset by software. Otherwise, it times out and causes a VCore-III soft reset equivalent to setting RESET.CORE_RST_FORCE. Improper use of the WDT.WDT_LOCK causes an immediate timeout-reset as if the watchdog timer had run out. The WDT.WDT_STATUS field shows if the last VCore-III CPU reset was caused by WDT timeout (or improper locking sequence). The WDT.WDT_STATUS field is updated only during VCore-III CPU reset.

To enable or to reset the watchdog timer, write the locking sequence, as described in WDT.WDT_LOCK, at the same time as setting the WDT.WDT_ENABLE field.

Because watchdog timeout is equivalent to setting RESET.CORE_RST_FORCE, the RESET.CORE_RST_CPU_ONLY field also applies to watchdog initiated soft reset.

## 5.3 Shared Bus

The following table lists the registers associated with the shared bus.

***Table 105 •*** **Shared Bus Configuration Registers**

| Register | Description |
|---|---|
| GENERAL_CTRL | Memory map and interface ownership configuration |
| PL1, PL2, PL3 | Master priorities |
| WT_EN | Weighted token scheme enable |
| WT_tcl | Weighted token refresh period |
| WT_CL1, WT_CL2, WT_CL3 | Token weights for masters |

The shared bus is a 32-bit address and 32-bit data bus with dedicated master and slave interfaces that interconnect all blocks in the VCore-III system. The VCore-III CPU, Frame DMA, and external CPU are masters on the shared bus and only they can start access on the bus.

The shared bus uses byte addresses, and transfers of 8, 16, or 32 bits can be made. For 16-bit and 32-bit access, the addresses must be aligned to 16-bit and 32-bit addresses, respectively. To increase performance, bursting of multiple 32-bit words on the shared bus can be performed.

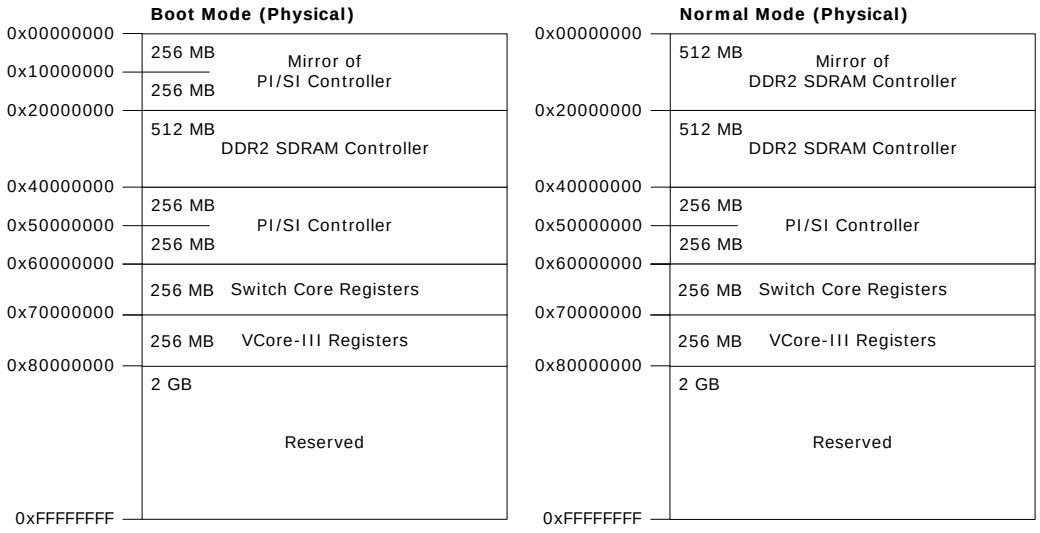
All slaves are mapped into the VCore-III systems 32-bit address space and can be accessed directly by masters on the shared bus. There are two possible mappings of VCore-III shared bus slaves:

- Boot mode. Boot mode is active after power-up and reset of the VCore-III system. In this mode, the PI and SI controller is mirrored into the lowest address region.
- Normal mode. In normal mode, the DDR2 SDRAM controller is mirrored into the lowest address region.

Changing from boot mode to normal mode (GENERAL_CTRL.BOOT_MODE_ENA) interchanges PI/SI for DDR2 SDRAM memory space.

The following illustration shows the mapping of the shared bus memory.

***Figure 38 •*** **Shared Bus Memory Map**




**Note** When the VCore-III system is protected from a soft reset using RESET.CORE_RST_CPU_ONLY, a soft reset or a watchdog timeout does not change shared bus memory mapping. For more information about protecting the VCore-III system when using a soft reset, see Clocking and Reset, page 133.

The SI interface is accessible through the lower 256 megabytes of the PI/SI controller's memory region. The upper 256 megabytes are reserved for the PI. The PI is mapped as overlaid functions on the GPIO interface. It is possible for the VCore-III CPU to take ownership of the PI interface by setting GENERAL_CTRL.IF_MASTER_PI_ENA, this automatically enables the parallel interface mode for the appropriate GPIO pins. For more information about the overlaid functions for the PI, seeOverlaid Functions on the GPIOs, page 177.

**Note** GENERAL_CTRL.IF_MASTER_PI_ENA must not be set when an external CPU is using the PI in slave mode for accessing the device.

In boot mode, the PI/SI controller's memory is mirrored into the lowest region of the memory map. In normal mode, the DDR2 SDRAM controller's memory is mirrored to the lowest region of the memory map. If the contents of the PI or SI memory and the DDR2 SDRAM memory are the same, software can execute from the mirrored region when swapping from boot mode to normal mode. Otherwise, software executes from the fixed PI/SI controller's memory when changing from boot mode to normal mode.

## 5.3.1 Shared Bus Arbitration

The VCore-III shared bus arbitrates between masters that want to access the bus; the default is to use a strict prioritized arbitration scheme where the VCore-III CPU has highest priority. Priorities can be changed using registers PL1 though PL3.

It is possible to enable weighted token arbitration scheme (WT_EN). When using this scheme, specific masters can be guaranteed a certain amount of bandwidth on the shared bus. Guaranteed bandwidth that is not used is given to other masters requesting the shared bus.

When weighted token arbitration is enabled, the masters on the shared bus are grated a configurable number of tokens (WT_CL1, WT_CL2, WT_CL3) at the start of each refresh period. The length of each refresh period is configurable (WT_TCL). For each clock-cycle that the master uses the shared bus, the token counter for that master is decremented. When all tokens are spent, the master is forced to a low priority. Masters with tokens always take priority over masters with no tokens. The strict prioritized scheme is used to arbitrate between masters with tokens and between masters without tokens.

Example: Guarantee That The Frame DMA Can Get 25% Bandwidth. Configure WT_TCL to a refresh period of 2048 clock cycles; the optimal length of the refresh period depends on the scenario, experiment to find the right setting. Guarantee Frame DMA access in 25% of the refresh period by setting WT_CL2

to 512 (2048 x 25%). Set WT_CL1 and WT_CL3 to 0. This gives the VCore-III CPU and External CPU unlimited tokens. Configure the Frame DMA to highest priority by setting PL2 to 15. Finally, enable the weighted token scheme by setting WT_EN to 1. For each refresh period of 2048 clock cycles, the Frame DMA is guaranteed access to the shared bus for 512 clock cycles, because it is the highest priority master. When all the tokens are spend, it is put into the low-priority category. Until the start of the next refresh period, the VCore-III CPU and the External CPU has priority when accessing the shared bus.

## 5.3.2 SI Memory Region

This section provides information about the functional aspects of the serial interface (SI) in master mode. For information about using an external CPU to access register targets using the serial interface, see Serial Interface in Slave Mode, page 159.

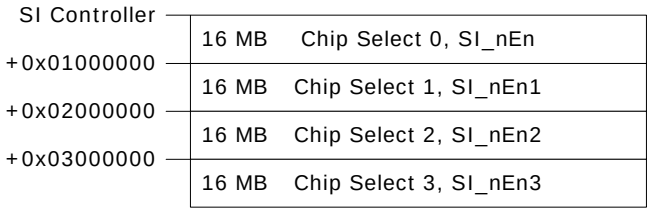
The following table lists the registers associated with the SI controller.

***Table 106 •*** **SI Controller Configuration Registers**

| Register | Description |
|---|---|
| SPI_MST_CFG | Serial interface speed |
| SW_MODE | Manual control of the serial interface pins |

When the VCore-III system controls the SI, there are four programmable chip selects. Through individually mapped memory regions, each chip select can address up to 16 megabytes of memory. Reading from the memory region for a specific SI chip select generates SI read on that chip select. It is possible for the VCore-III CPU to execute code directly from Flash by executing from the SI Controller's memory region.

***Figure 39 •*** **SI Controller Memory Map**




The SI controller accepts 8-bit, 16-bit, and 32-bit read-access with or without bursting. Writing to the SI requires manual control of the SI-pins using software. Setting SW_MODE.SW_PIN_CTRL_MODE places all SI pins under software control. Output enable and the value of SI_Clk, SI_DO, SI_nEn[3:0] are controlled through the SW_MODE register. The value of the SI_DI pin is available in SW_MODE.SW_SPI_SDI.

**Note** The VCore-III CPU cannot execute code directly from the SI controller's memory region at the same time as manually writing to the serial interface.

The following table lists the serial interface pins.

***Table 107 •*** **Serial Interface Pins**

| Pin Name | I/O | Description |
|---|---|---|
| SI_nEN<br>SI_nEN1, GPIO<br>SI_nEN2, GPIO<br>SI_nEN3, GPIO | O | Active low chip selects. Only one chip select can be active at any time. Chip selects 1 through 3 are overlaid functions on the GPIOs. See Overlaid Functions on the GPIOs, page 177. |
| SI_Clk | O | Clock output. |
| SI_DO | O | Data output (MOSI). |
| SI_DI | I | Data input (MISO). |

The SI controller does speculative perfecting of data. After reading address *n*, the SI controller automatically continues reading address *n* + 1, so that the next value is ready if or when requested by the VCore-III CPU. This greatly optimizes reading from sequential addresses in the Flash, such as when copying data from Flash into program memory.

***Figure 40 •*** **SI Read Timing in Normal Mode**

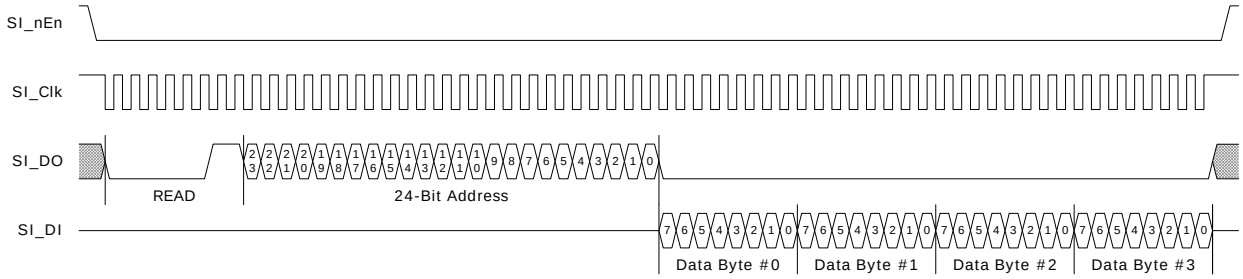
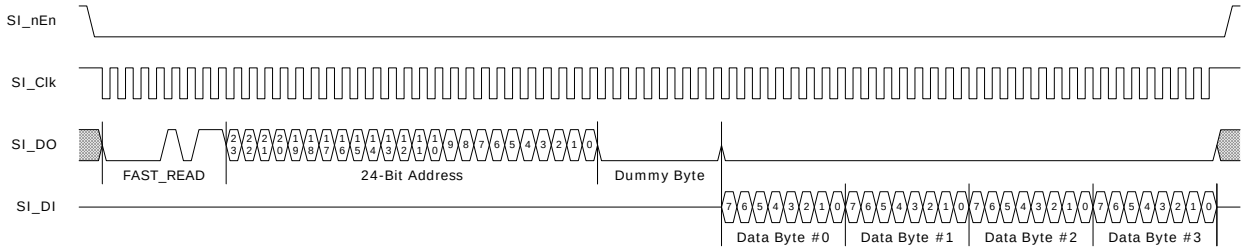



***Figure 41 •*** **SI Read Timing in Fast Mode**




The default timing of the SI controller operates with most serial interface Flash devices. Use the following process to calculate the optimized SI parameters for a specific SI device:

1. Calculate an appropriate frequency divider value as described in SPI_MST_CFG.CLK_DIV. The SI operates at no more than 25 MHz, and the maximum frequency of the SPI device must not be exceeded. For information about the VCore-III system frequency, see Clocking and Reset, page 133.
2. The SPI device may require a FAST_READ command rather than normal READ when the SI frequency is increased. Setting SPI_MST_CFG.FAST_READ_ENA makes the SI controller use FAST_READ commands.
3. Calculate SPI_MST_CFG.CS_DESELECT_TIME so that it matches how long the SPI device requires chip-select to be deasserted between accesses. This value depends on the SI clock period that results from the SPI_MST_CFG.CLK_DIV setting.

These parameters must be written to SPI_MST_CFG. The CLK_DIV field must either be written last or at the same time as the other parameters. The SPI_MST_CFG register can be configured while also booting up from the SI.

When the VCore CPU boots from the SI interface, the default values of the SPI_MST_CFG register are used until the SI_MST_CFG is reconfigured with optimized parameters. This implies that SI_Clk is operating at approximately 4 MHz, with normal read instructions, and maximum gap between chip select operations to the Flash.

## 5.3.3 PI Memory Region

This section provides information about the functions of the parallel interface (PI) in master mode. For information about how an external CPU can access register targets using the PI, see Parallel Interface in Slave Mode, page 161.

The following table lists the PI controller registers.

***Table 108 •*** **PI Controller Configuration Registers**

| Pin Name | Description |
|---|---|
| PI_MSI_CFG | Parallel interface speed |
| PI_MST_CTRL | Configuration of interface width, transfer type, and timing |
| PI_MST_STATUS | Timeout indication |
| GENERAL_CTRL | Enables the PI master |

The parallel interface on the device is optimized for NAND Flash access and for connection to external programmable logic. There are four address pins available. The PI chip select is mapped to the low part of the PI controller memory region. There are no limitations on the type of access that can be done within this region; 8-bit, 16-bit, and 32-bit access with or without bursting are all translated to an appropriate number of accesses on the parallel interface.

The parallel interface pins on the device are all overlaid functions on the GPIO interface. Before accessing the parallel interface, the VCore-III system must take ownership of the PI using GENERAL_CTRL.IF_MASTER_PI_ENA. which automatically overtakes the appropriate GPIO pins. For more information, see Overlaid Functions on the GPIOs, page 177.

The following table lists the parallel interface pins.

***Table 109 •*** **Parallel Interface Pins**

| Pin Name | I/O | Description |
|---|---|---|
| PI_nCS, GPIO | O | Active low chip selects. Only one chip select can be active at any time. |
| PI_Addr[3:0], GPIO | O | These are the address lines. The least significant bit is 0, and the most significant bit is 25. |
| PI_nWR, GPIO | O | Active low write enable. This is asserted throughout write access on the PI. |
| PI_nOE, GPIO | O | Active low output enable. This is asserted during read access on the parallel interface. |
| PI_Data[7:0], GPIO | I/O | These are the data lines. |
| PI_nDone, GPIO | I | An external device can use this input to indicate when a transfer is done. This input is only used when a chip select is configured to use device-paced mode. See Device-Paced Mode, page 139. |

The timing of the parallel interface is described in clock cycles. This refers to PI_Clk, which is a clock derived from the VCore-III system clock (PI_MST_CFG.CLK_DIV). In the PI controller, all signals are set or sampled on the rising edge of PI_Clk.

Successive accesses on PI are always spaced with at least one PI_Clk cycle. However, when an access to the PI controller is wider than the interface (for example, 32-bit access to an 8-bit interface), the access is split into multiple back-to-back access.

For read and write access, there are three functional timing parameters that can be adjusted.

- CSCC: The delay from setting PI_Addr, Pi_nWR, PI_nOE, and PI_nBE until PI_nCS is asserted.
- WAITCC+1: The delay from starting an access to PI_nCS is deasserted.
- HLDCC: The delay from deasserting PI_nCS until control signals are changed.

For read access, one additional parameter applies:

- OECC: The delay from PI_nCS is asserted to PI_nOE is asserted.

For read access, data is sampled at the same time as PI_nCS is deasserted.

The following illustrations show the PI write and read timing. The internal PI_Clk signal is included to illustrate the functional PI timing.

***Figure 42 •*** **PI Write Timing**

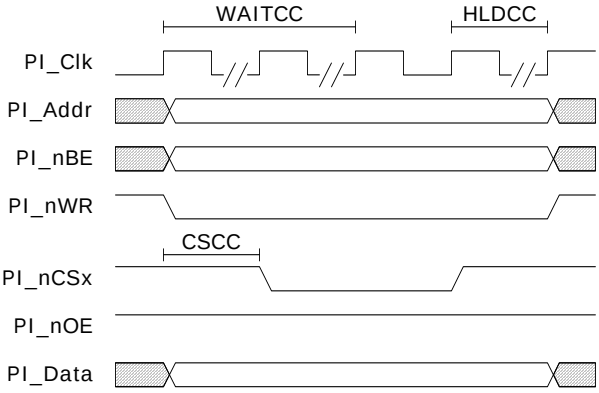
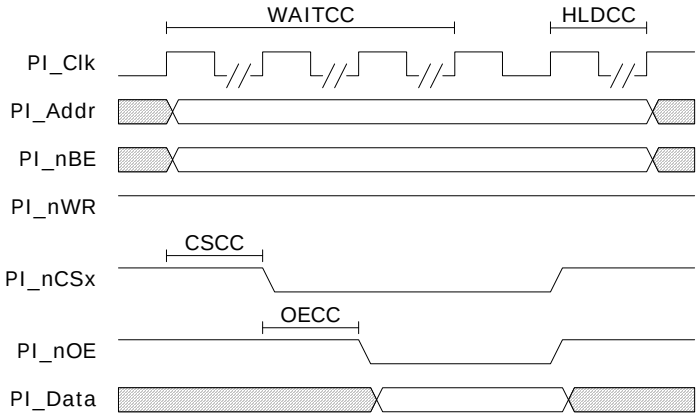



***Figure 43 •*** **PI Read Timing**



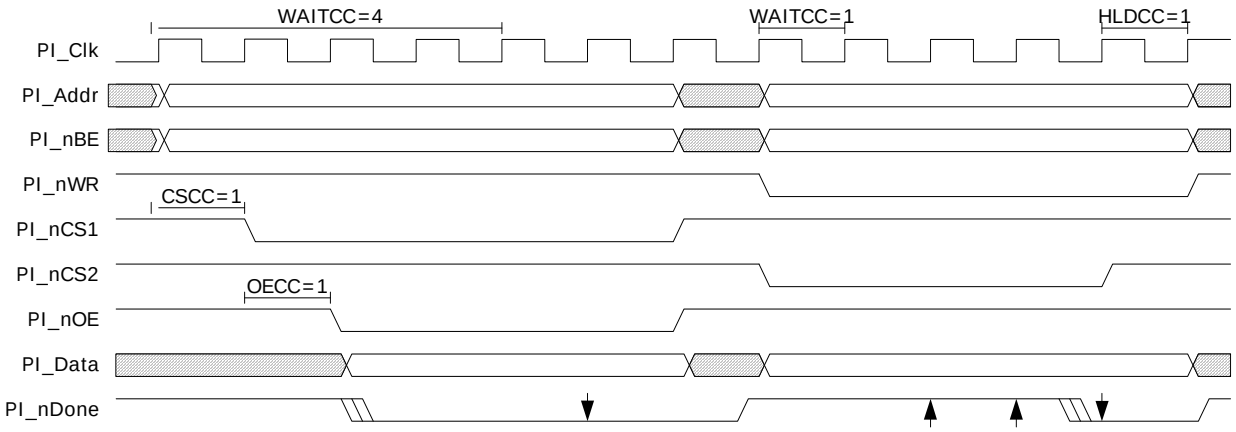

For both read and write access, WAITCC must be greater than or equal to CSCC + OECC. The WAITCC, CSCC, and OECC parameters can be zero, as well as HLDCC. If all parameters are zero, access is done in a single PI clock cycle.

#### 5.3.3.1 Device-Paced Mode

Device-paced mode can be enabled using PI_MST_CTRL.DEVICE_PACED_XFER_ENA. When device-paced mode is enabled, the cycle in-between WAITCC and HLDCC is stretched until an external device allows the access to be completed by signaling on the PI_nDone pin. The default polarity of this signal is active low, but it can be changed (PI_MST_CTRL.DONE_POL). The PI controller starts to sample the PI_nDone pin after the WAITCC part is over. After the PI_nDone signal is asserted, the PI controller waits one additional cycle. It then proceeds with the transfer (and sample data when reading) by going into the hold-period, or terminates the transfer if HLDCC is zero. The one cycle delay after detecting an asserted PI_nDone signal can be removed, allowing the PI controller to read data and proceed in the same cycle as PI_nDone is detected (PI_MST_CTRL.SAMPLE_ON_DONE).

Example: Use of Device-Paced Mode  This example shows two different configurations of chip selects; called PI_nCS1 and PI_nCS2, the example shows a read using PI_nCS1 and a write using PI_nCS2. PI_nCS1 is configured with CSCC = 1, OECC = 1, WAITCC = 4, and HLDCC = 0. PI_nCS2 is configured with CSCC = 0, WAITCC = 1, and HLDCC = 1. Both configurations have device-paced mode enabled, and PI_nCS2 is configured with SAMPLE_ON_DONE.

***Figure 44 •*** **Device-Paced PI Example**




The arrows show where the PI controller samples PI_nDone. Note how PI_MST_CTRL.SAMPLE_ON_DONE causes the access on PI_nCS2 to proceed in the same cycle as PI_nDone is asserted, as opposed to PI_nCS1.

PI_nDone is an asynchronous signal. It takes a maximum of two VCore-III system clock cycles for the PI controller to detect an asserted PI_nDone signal.

In device-paced mode, a timeout can be enabled using PI_MST_CTRL.DEVICE_PACED_TIMEOUT_ENA. The timeout period counts from the start of the access and is configured in the range of 16 through 2048 cycles (PI_MST_CTRL.DEVICE_PACED_TIMEOUT). If a timeout occurs, a transfer is immediately terminated; reads return invalid data. The sticky register bit PI_MST_STATUS.TIMEOUT_ERR_STICKY is set when a transfer has timed out.

## 5.3.4 DDR2 Memory Region

This section provides information about how to configure the DDR2 memory controller and interface.

The following table lists the registers associated with the DDR2 controller.

***Table 110 •*** **DDR2 Controller Registers**

| Register | Description |
|---|---|
| MEMCTRL_CTRL | Start of initialization |
| MEMCTRL_CFG | Configuration |
| MEMCTRL_STAT | Status for initialization |
| MEMCTRL_REF_PERIOD | Refresh period |
| MEMCTRL_TIMING0 | Timing configuration |
| MEMCTRL_TIMING1 | Timing configuration |
| MEMCTRL_TIMING2 | Timing configuration |
| MEMCTRL_TIMING3 | Timing configuration |
| MEMCTRL_MR0_VAL | Mode register 0 value |
| MEMCTRL_MR1_VAL | Mode register 1 value |
| MEMCTRL_MR2_VAL | Mode register 2 value |
| MEMCTRL_MR3_VAL | Mode register 3 value |
| MEMCTRL_DQS_DLY | DQS window configuration |

*Table 110 •* **DDR2 Controller Registers *(continued)***

| Register | Description |
|---|---|
| MEMPHY_CFG | Interface configuration |
| MEMPHY_ZCAL | Interface calibration |

The memory controller is designed to work with JEDEC-compliant DDR2 memory modules. The controller supports up to 14 addresses, 4 or 8 bank, and single row configurations (fixed CS). The memory controller has a single byte lane supporting one 8-bit DDR2 module.

**Note** The memory controller supports single row systems, which means there is no DDR_nCS output; the nCS input on the DDR2 module must be tied to 0.

The following steps are required to bring up the memory controller:

1. Configure timing and mode parameters. Configuration depends on the DDR2 modules selected for the product. For more information, see Configuration of Timing and Mode Parameters, page 141.
2. Enable and calibrate the SSTL I/Os. For more information, see Enabling and Calibrating the SSTL I/Os, page 143.
3. Initialize the memory controller and modules. For more information, see Memory Controller and Module Initialization, page 144.
4. Calibrate the DQS read window. For more information, DQS Read Window Calibration, page 144.

**Note** For selected DDR2 modules, the bring-up of the memory controller is already implemented as part of the Board Support Package (BSP). Please see the BSP for example implementation of the bring-up procedure.

### 5.3.4.1 Configuration of Timing and Mode Parameters

This section lists each of the parameters that must be configured prior to initialization of the memory controller. The register list contains a more comprehensive explanation of each field; this section provides a quick overview of fields that must be configured and the recommended values.

All divisions in this section are performed as floating point division and then rounded up to nearest integer, unless otherwise is explicitly mentioned for that division.

The following table defines the variables that must be extracted from the datasheet of the DDR2 module (referred to as “module”) that have selected for use with the device. Note that some of the variables listed in the table depend on the frequency at which the module is run. It is assumed that a target frequency was determined. For more information, see Clocking and Reset, page 133.

*Table 111 •* **Selected Memory Module Variables**

| Variable | Description |
|---|---|
| clk_ns | The clock period in nanoseconds at which the module runs. |
| CL | The CAS latency of the module in clock cycles. |
| $t_{REFI_ns}$ | The $t_{REFI}$ parameter for the module in nanoseconds. |
| $t_{WR_ns}$ | The $t_{WR}$ parameter for the module in nanoseconds. |
| $t_{RAS_min_ns}$ | The $t_{RAS(MIN)}$ parameter for the module in nanoseconds. |
| $t_{WTR_ns}$ | The $t_{WTR}$ parameter for the module in nanoseconds. |
| $t_{RCD_ns}$ | The $t_{RCD}$ parameter for the module in nanoseconds. |
| $t_{RRD_ns}$ | The $t_{RRD}$ parameter for the module in nanoseconds. |
| $t_{RP_ns}$ | The $t_{RP}$ parameter for the module in nanoseconds. |
| $t_{FAW_ns}$ | The $t_{FAW}$ parameter for the module in nanoseconds. Required for 8-bank modules. |
| $t_{RC_ns}$ | The $t_{RC}$ parameter for the module in nanoseconds. |
| $t_{RFC_ns}$ | The $t_{RFC}$ parameter for the module in nanoseconds. |

*Table 111 •* **Selected Memory Module Variables *(continued)***

| Variable | Description |
|---|---|
| $t_{MRD}$ | The $t_{MRD}$ parameter for the module in clock cycles. |
| $t_{RPA_ns}$ | The $t_{RPA}$ parameter in nanoseconds. Required for 8-bank modules. |

The timing parameters listed in the following table must be configured. For more information about each register field, see the detailed register on each field. Where multiple configurations are possible, the most optimal solution is selected.

*Table 112 •* **Memory Controller Timing Parameters**

| Timing Parameter | Description |
|---|---|
| MEMCTRL_CFG.MSB_COL_ADDR | Set to one less than the number of column address bits for the DDR2 module. |
| MEMCTRL_CFG.MSB_ROW_ADDR | Set to one less than the number of row address bits for the DDR2 module. |
| MEMCTRL_CFG.BANK_CNT | Set to 0 when using a 4-bank DDR2 module. Set to 1 when using an 8-bank DDR2 module. |
| MEMCTRL_CFG.BURST_LEN | Set to 1, BURST8 mode. |
| MEMCTRL_CFG.BURST_SIZE | Set to 0. |
| MEMCTRL_REF_PERIOD.REF_PERIOD | Set to (tREFI_ns/clk_ns). Round down the result to the nearest integer. |
| MEMCTRL_REF_PERIOD.MAX_PEND_REF | Set to 1. |
| MEMCTRL_TIMING0.RD_DATA_XFR_DLY | Set to (CL – 3). |
| MEMCTRL_TIMING0.WR_DATA_XFR_DLY | Set to (CL – 3). |
| MEMCTRL_TIMING0.RD_TO_PRECH_DLY | Set to 3. |
| MEMCTRL_TIMING0.WR_TO_PRECH_DLY | Set to (CL + 2 + (tWR_ns/clk_ns)). |
| MEMCTRL_TIMING0.RAS_TO_PRECH_DLY | Set to ((tRAS_min_ns/clk_ns) – 1). |
| MEMCTRL_TIMING0.RD_TO_WR_DLY | Set to 4. |
| MEMCTRL_TIMING1.WR_TO_RD_DLY | Set to the highest value of either (CL + 4) or (CL + 2 + (tWTR_ns/clk_ns)). |
| MEMCTRL_TIMING1.RAS_TO_CAS_DLY | Set to ((tRCD_ns/clk_ns) – 1). |
| MEMCTRL_TIMING1.RAS_TO_RAS_DLY | Set to ((tRRD_ns/clk_ns) – 1). |
| MEMCTRL_TIMING1.PRECH_TO_RAS_DLY | Set to ((tRP_ns/clk_ns) – 1). |
| MEMCTRL_TIMING1.BANK8_FAW_DLY | Set to 0 for a 4-bank module. Set to ((tFAW_ns/clk_ns) – 1) for an 8-bank module. |
| MEMCTRL_TIMING1.RAS_TO_RAS_SAME_BANK_DLY | Set to ((tRC_ns/clk_ns) – 1). |
| MEMCTRL_TIMING2.FOUR_HUNDRED_NS_DLY | Set to (400 /clk_ns). |
| MEMCTRL_TIMING2.REF_DLY | Set to ((tRFC_ns/clk_ns) – 1). |
| MEMCTRL_TIMING2.MDSET_DLY | Set to (tMRD – 1). |
| MEMCTRL_TIMING2.PRECH_ALL_DLY | Set to ((tRP_ns/clk_ns) – 1) for a 4-bank module. Set to ((tRPA_ns/clk_ns) – 1) for an 8-bank module. |

*Table 112 •* **Memory Controller Timing Parameters *(continued)***

| Timing Parameter | Description |
|---|---|
| MEMCTRL_TIMING3.WR_TO_RD_CS_CHANGE_DLY | Set to the highest value of either 3 or (CL – 1). |
| MEMCTRL_TIMING3.LOCAL_ODT_RD_DLY | Set to (CL – 1). |
| MEMCTRL_TIMING3.ODT_WR_DLY | Set to (CL – 1). |

The memory controller supports single-row systems, which implies that the data connections between the memory controller and the DDR2 modules are point-to-point connections. As a result, on-die-termination is not required.

The following table lists the mode parameters that need to be configured. The suggestions in the table are inline with the timing parameters that are listed in the previous table. Where multiple configurations are possible, the most optimal solution is selected.

*Table 113 •* **Memory Controller Mode Parameters**

| Mode Parameter | Description |
|---|---|
| MEMCTRL_MR0_VAL.MR0_VAL | This value is written to the Mode register in the DDR2 module during initialization. Set to (3\|(CL<<4)\|(((tWR_ns/clk_ns) – 1)<<9)) |
| MEMCTRL_MR1_VAL.MR1_VAL | This value is written to the Extended Mode register in the DDR2 module during initialization. Set to 0x0382. |
| MEMCTRL_MR2_VAL.MR2_VAL | This value is written to the Extended Mode Register 2 in the DDR2 module during initialization. Set to 0x0000. |
| MEMCTRL_MR3_VAL.MR3_VAL | This value is written to the Extended Mode Register 3 in the DDR2 module during initialization. Set to 0x0000. |

The mode registers are specified by the JEDEC standards, and bit positions in the mode registers across different DDR2 vendors remain fixed.

#### 5.3.4.2 Enabling and Calibrating the SSTL I/Os

The memory controller is designed to operate with point-to-point PCB traces on the timing critical control and data connections to and from the DDR2 modules.

Prior to controller initialization, the device's SSTL I/O drivers must be enabled and calibrated to correct drive strength and termination resistor values. For single row systems with short point-to-point connections, it is recommended that the device's I/O drive strength be 60 Ω/60 Ω. Using these values ensures proper low power and low noise communication.

Complete the following tasks to enable and calibrate the SSTL I/Os:

1. Release the I/Os and related logic from reset.
2. Enable the SSTL mode by clearing MEMPHY_CFG.PHY_RST and setting MEMPHY_CFG.PHY_SSTL_ENA.
3. Perform calibration with the previously mentioned strength and termination values by writing 0xEH to MEMPHY_ZCAL.
4. Ensure that software waits until MEMPHY_ZCAL.ZCAL_ENA is cleared (indicates calibration is done) before continuing.
5. Enable drive of the SSTL I/Os by setting MEMPHY_CFG.PHY_CK_OE, MEMPHY_CFG.PHY_CL_OE, and MEMPHY_CFG.PHY_ODT_OE.

The SSTL interface is now enabled and calibrated, and the initialization of the memory controller can commence.

#### 5.3.4.3 Memory Controller and Module Initialization

After all timing parameters and mode registers are configured, and after the SSTL I/Os are enabled and calibrated, the memory controller (and DDR2 modules) can be initialized by setting MEMCTRL_CTRL.INITIALIZE. For more information about configuring timing and mode parameters, see Configuration of Timing and Mode Parameters, page 141. For more information about the DDR2 SSTL I/Os, see Enabling and Calibrating the SSTL I/Os, page 143.

During initialization, the memory controller automatically follows the proper JEDEC defined procedure for initialization and writing of mode registers to the DDR2 memory modules.

The memory controller sets the MEMCTRL_STAT.INIT_DONE field after the controller and the DDR2 memory are operational. Software must wait for the INIT_DONE indication before continuing to calibrate the read window.

#### 5.3.4.4 DQS Read Window Calibration

After initialization of the memory controller, writes to the memory are guaranteed to be successful. Reading is not yet possible, however, because the round trip delay between controller and DDR2 modules is not calibrated.

Calibration of the read window includes writing a known value to the start of the DDR memory and then continually reading this value while adjusting the DQS window until the correct value is read from the memory.

Complete the following steps before starting the calibration routine:

- Write 0x000000FF to SBA address 0x20000000
- Set the MEMCTRL_DQS_DLY.DQS_DLY field to 0.

Perform the following steps to calibrate the read window. Do not increment the DQS_DLY field beyond its maximum value. If the DQS_DLY maximum value is exceeded, it is an indication something is incorrect, and the DDR2 memory will not be functional.

1. Read byte address 0 from the DDR2 memory. If the content of byte address 0 is different from 0xFF, increment MEMCTRL_DQS_DLY.DQS_DLY by one, and repeat step 1, else continue to step 2.
2. Read byte address 0 from the DDR2 memory. If the content of byte address 0 is different from 0x00, increment MEMCTRL_DQS_DLY.DQS_DLY by one, and repeat step 2, else continue to step 3.
3. Decrement MEMCTRL_DQS_DLY.DQS_DLY by three.

The last step configures the appropriate DSQ read window. The DDR memory is operational after this step and can be used for random access.

### 5.3.5 Switch Core Registers Memory Region

Register targets in the Switch Core are memory-mapped into the Switch Core registers region of the shared bus memory map. All registers are 32-bit wide and must only be accessed using 32-bit reads and writes. Bursts are supported.

Writes to this region are buffered (there is a one-word write buffer). Multiple back-to-back write access pauses the shared bus until the write buffer is freed up (until the previous writes are done). Reads from this region pause the shared bus until read data is available.

Registers in the 0x60000000 though 0x6FFFFFFF region in the 0x6 targets are physically located in other areas of the device rather than the VCore-III system; reading from these targets may take up to 1.1 µs in a single master system. For more information, see Register Access and Multimaster Systems, page 159.

### 5.3.6 VCore-III Registers Memory Region

Registers inside the VCore-III domain are memory mapped into the VCore-III registers region of the shared bus memory map. All registers are 32-bit wide and must only be accessed using 32-bit reads and writes, bursts are supported.

The registers in the 0x70000000 though 0x7FFFFFFF region are all placed inside the VCore-III, read and write access to these registers is fast (done in a few clock cycles).

## 5.4 VCore-III CPU

The VCore-III CPU system is based on a powerful MIPS24KEc-compatible microprocessor with 16-entry MMU, 32 kilobyte instruction, and 32 kilobyte data caches.

This section describes how the VCore-III CPU is integrated into the VCore-III system. For more information about internal VCore-III functions, for example, bringing up caches, MMU, and so on.

When automatic boot is enabled using the VCORE_CFG strapping pins, the VCore-III CPU automatically starts to execute code in the Flash at byte-address 0.

A typical automatic boot sequence is as follows:

1. Configure appropriate VCore-III CPU frequency. For more information, see Clocking and Reset, page 133.
2. Speed up the boot interface. For more information, see Shared Bus, page 134.
3. Initialize the DDR2 controller and memory. For more information, see DDR2 Memory Region, page 140.
4. Copy code-image from Flash to DDR2 memory.
5. Change memory map from boot mode to normal mode For more information, see Shared Bus, page 134.

When automatic boot is disabled, an external CPU can start the VCore-III CPU through registers.

A typical manual boot sequence is:

1. Configure appropriate VCore-III CPU frequency. For more information, see Clocking and Reset, page 133.
2. Initialize the DDR2 controller and memory. For more information, see DDR2 Memory Region, page 140.
3. Copy code-image to DDR2 memory.
4. Change memory map from boot mode to normal mode. For more information, see Shared Bus, page 134.
5. Release reset to the VCore-III CPU. For more information, see Clocking and Reset, page 133.

The boot vector of the VCore-III CPU is mapped to the start of the KESEG1, which translates to physical address 0x00000000 on the VCore-III shared bus.

The VCore-III interrupts are mapped to interrupt inputs 0 and 1, respectively.

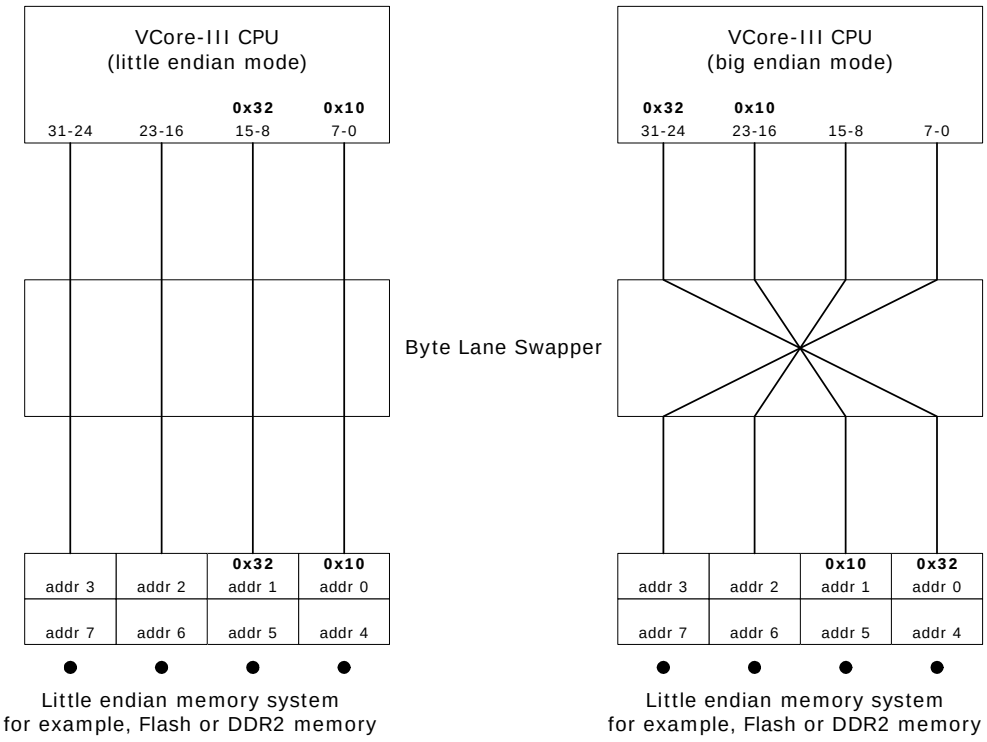
### 5.4.1 Big Endian Support

The endianess of the VCore-III CPU is controlled through strapping pins. For more information about how to select endian modes, see VCore-III System and CPU Interface, page 131.

The VCore-III system is constructed as a little endian system, and registers descriptions reflect little endian encoding. When big endian mode is enabled, instructions and data are byte-lane swapped just before they enter and when they leave the VCore-III CPU. This is the standard way of translating between a CPU in big endian mode and a little endian system.

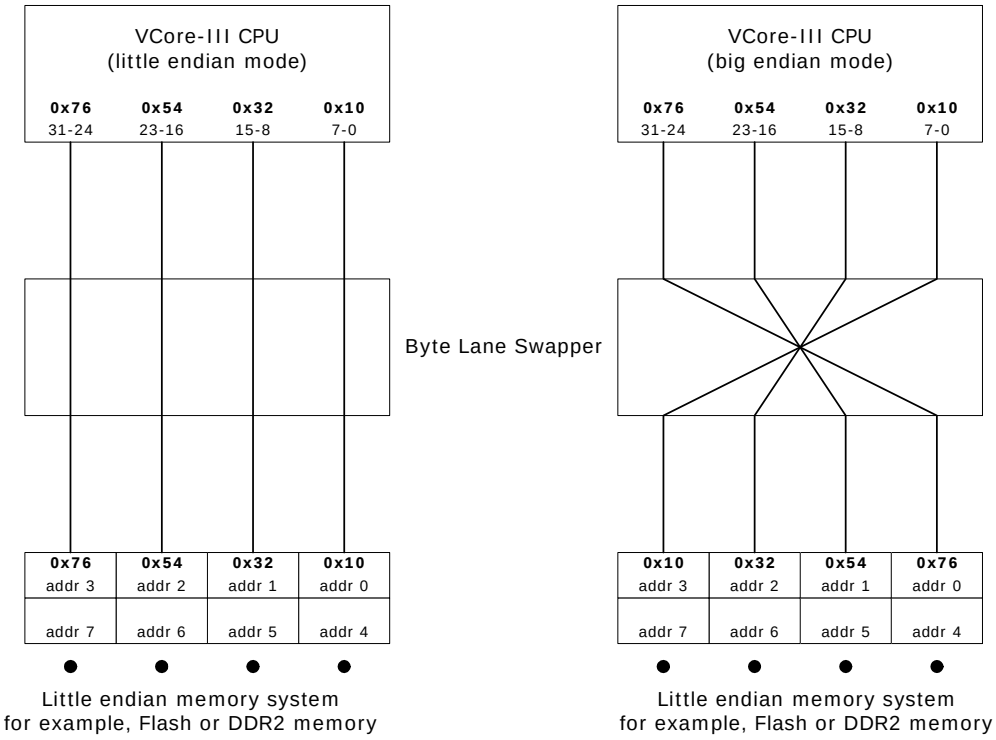
The following illustration shows how the 16-bit value 0x3210 is transferred between the VCore-III CPU and the VCore-III shared bus in little endian and big endian modes.

***Figure 45 •*** **16-Bit Access in Little Endian and Big Endian Modes**




For 32-bit access, the difference is less obvious. The following illustration shows how the value 0x76543210 is transferred between the VCore-III CPU and the VCore-III shared bus in little endian and big endian modes.

***Figure 46 •*** **32-Bit Access in Little Endian and Big Endian Mode**



**Note** The swapping of byte lanes ensures that no matter the endian mode, the VCore-III CPU is always accessing the appropriate part of the little endian memory system.

In big-endian mode, care must be taken when accessing parts of the memory system which is also used by other users than the VCore-III CPU. For example, device registers are written and read by the VCore-III CPU, but they are also used by the device (which sees them in little endian mode). The VCore-III BSP contains examples of code that correctly handles register access for big endian mode.

### 5.4.2 Software Debug and Development

The VCore-III CPU has a standard MIPS EJTAG debug interface that can be used for breakpoints, loading of code, and examining memory. When the VCore_ICE_nEn strapping pin is pulled low, the device's JTAG interface is attached to the VCore-III EJTAG controller.

## 5.5 Manual Frame Injection and Extraction

This section provides information about the manual frame injection and extraction to and from the CPU system. The devices have two injection groups and two extraction groups available.

### 5.5.1 Manual Frame Extraction

This section provides information about manual frame extraction.

The following table lists the registers associated with manual frame extraction.

***Table 114 •*** **Manual Frame Extraction Registers**

| Register | Description | Replication |
|---|---|---|
| XTR_FRM_PRUNING | Frame pruning | Per xtr queue |
| XTR_GRP_CFG | Extraction group configuration | Per xtr group |
| XTR_MAP | Map extraction queue to group | Per xtr queue |
| XTR_RD | Extraction read data | Per xtr group |
| XTR_QU_SEL | Software controlled queue selection | Per xtr group |
| XTR_QU_FLUSH | Extraction queue flush | None |
| XTR_DATA_PRESENT | Extraction status | None |

The devices have two extraction queues to which data can be redirected. Before data can be extracted each extraction queue must be enabled and mapped to an extraction group. The devices have two extraction groups available, and the mapping between queues and groups can be set arbitrary. A queue is enabled by setting the corresponding XTR_MAP.MAP_ENA field and the mapping to an extraction group is set in XTR_MAP.GRP.

The XTR_DATA_PRESENT register shows if data is present in the extraction queues. It has two fields:

- XTR_DATA_PRESENT.DATA_PRESENT shows the data present status per extraction queue
- XTR_DATA_PRESENT.DATA_PRESENT_GRP shows the data present status per extraction group.

When frame data is available in an extraction group, it can be read from the associated XTR_RD register, which is replicated per extraction group. The XTR_RD register returns the next 4 bytes of the frame data. When the read operation is completed, the register is automatically updated with the next 4 bytes of the frame data. End-of-frame (EOF) and other status indications are indicated by special data words in the data stream (when reading XTR_RD). The following table lists the possible special data words.

***Table 115 •*** **Extraction Data Special Values**

| Data Value | Description |
|---|---|
| 0x80000000-0x80000003 | EOF. The two LSBs indicate the number of unused bytes. |
| 0x80000004 | EOF. Frame was pruned. |

*Table 115 •* **Extraction Data Special Values *(continued)***

| Data Value | Description |
|---|---|
| 0x80000005 | EOF. The frame was aborted and is invalid. |
| 0x80000006 | Escape. Next data is frame data and not a status word. |
| 0x80000007 | Data not ready. |

Each read operation on the XTR_RD register must check for the special values listed above and act accordingly. The escape data word (0x80000006) is inserted into the data stream when the frame data matches one of the special data words. When the escape data word is read it means that the next data word to be read is actual frame data and not a status word.

The position of the EOF data word in the data stream can be configured in XTR_GRP_CFG.STATUS_WORD_POS. The possibilities are to have the EOF status word after the last frame data word or to have EOF status word just before the last frame data word. The default is to have the EOF status word after the last frame data word.

The byte order of the XTR_RD register can be configured in XTR_GRP_CFG.BYTE_SWAP. The default is to have the byte order in little-endian. By clearing XTR_GRP_CFG.BYTE_SWAP, the byte order is changed to big-endian (network order). The byte order of the status words listed in Table 115, page 147 is not affected by the value of XTR_GRP_CFG.BYTE_SWAP.

It is possible to configure a prune size for all extracted frames from an extraction queue using XTR_FRM_PRUNING. When pruning is enabled, all frames that are larger than the specified prune size is pruned to the prune size. When a frame is pruned, the EOF status word is set to 0x80000004. The maximum prune size is 1024 bytes, and the prune size is defined in whole 32-bit words only.

Frames in individual extraction queues can be flushed by setting the corresponding bit in XTR_QU_FLUSH.FLUSH. Flushing is disabled by clearing XTR_QU_FLUSH.FLUSH.

**Note** Flushing does not affect the queues in the OQS so it may be needed to make the OQS stop sending data to the CPU extraction queues before flushing.

When a frame is extracted, it can be prefixed with an 8-byte CPU extraction header (EH). The option to prefix an EH to the frame data is set in the rewriter. For more information about the extraction header format, see CPU Extraction Header, page 121.

The extraction queue from which the frame originates is available through the CPU_QUEUE field in the CPU extraction header.

The following table shows an example of reading a 65-byte frame, followed by a 64-byte frame. In the example, it is assumed that each frame is prefixed with an EH. Data is read big endian, and the EOF status word is configured to come just before the last frame data word. Undefined bytes cannot be assumed to be zero.

*Table 116 •* **Frame Extraction Example**

| Read Number | INJ_WR Bits 31:24 | INJ_WR Bits 23:16 | INJ_WR Bits 15:8 | INJ_WR Bits 7:0 |
|---|---|---|---|---|
| 1 | EH bit 63:56 | EH bit 55:48 | EH bit 47:40 | EH bit 39:32 |
| 2 | EH bit 31:24 | EH bit 23:16 | EH bit 15:8 | EH bit 7:0 |
| 3 | Frame byte 1 (DMAC) | Frame byte 2 (DMAC) | Frame byte 3 (DMAC) | Frame byte 4 (DMAC) |
| 4 | Frame byte 5 (DMAC) | Frame byte 6 (DMAC) | Frame byte 7 (SMAC) | Frame byte 8 (SMAC) |
| … | | | | |
| 19 | 0x80 (EOF) | 0x00 (EOF) | 0x00 (EOF) | 0x03 (EOF) |

*Table 116 •* **Frame Extraction Example *(continued)***

| Read Number | INJ_WR Bits 31:24 | INJ_WR Bits 23:16 | INJ_WR Bits 15:8 | INJ_WR Bits 7:0 |
|---|---|---|---|---|
| 20 | Frame byte 65 (FCS) | Undefined | Undefined | Undefined |
| 21 | EH bit 63:56 | EH bit 55:48 | EH bit 47:40 | EH bit 39:32 |
| … | | | | |
| 38 | 0x80 (EOF) | 0x00 (EOF) | 0x00 (EOF) | 0x00 (EOF) |
| 39 | Frame byte 61 | Frame byte 62 | Frame byte 63 | Frame byte 64 |

## 5.5.2 Manual Frame Injection

This section provides information about manual frame injection on the devices.

The following table lists the register associated with manual frame injection.

*Table 117 •* **Manual Frame Injection Registers**

| Register | Description | Replication |
|---|---|---|
| INJ_GRP_CFG | Injection group configuration | Per injection group |
| INJ_WR | Injection write data | Per injection group |
| INJ_CTRL | Injection control | Per injection group |
| INJ_STATUS | Injection status | None |
| INJ_ERR | Injection errors | Per injection group |

The devices have two injection groups available. Frames can be injected from the CPU injection groups using register writes. There are two ways of injecting frames:

- Directly forwarding to a specific port, bypassing the analyzer.
- Normal forwarding of a frame through the analyzer.

To control the injection mode, an 8-byte injection header (IH) must be prefixed to the frame data. For more information about the injection modes and the injection header, see Frame Injection, page 122.

Frame data is injected by doing consecutive writes of 4 bytes to the INJ_WR register, which is replicated per injection group. Endianess of the INJ_WR register is configured in INJ_GRP_CFG.BYTE_SWAP. Start-of-frame (SOF) and end-of-frame (EOF) indications are set in INJ_CTRL. INJ_CTRL must be written prior to INJ_WR. SOF and EOF is indicated in INJ_CTRL.SOF and INJ_CTRL.EOF respectively. In INJ_CTRL.VLD_BYTES the number of valid bytes of the last write to INJ_WR is indicated and VLD_BYTES must be set together with the EOF indication. The frame data must include the 4-byte FCS, but it does not have to be correct, because it is recalculated by the egress port module. While a frame is being injected it can be aborted by setting INJ_CTRL.ABORT. The SOF, EOF, and ABORT fields of INJ_CTRL are automatically cleared by hardware.

Dummy bytes can be injected in front of a frame before the actual frame data (including injection header). The dummy bytes are discarded before the frame data is transmitted by the CPU system. The number of bytes to discard from the frame data is set in INJ_CTRL.GAP_SIZE. The GAP_SIZE field must be set together with SOF.

Before each write to INJ_WR, the status fields INJ_STATUS.WMARK_REACHED and INJ_STATUS.FIFO_RDY must be checked to ensure successful injection. The INJ_ERR register shows if an error occurred during frame injection.

The following table shows an example of injecting a 65-byte frame followed by a 64-byte frame. Both frames are prefixed by a CPU injection header and big-endian mode is used for the INJ_WR register. The “don’t care” bytes can be any value.

***Table 118 •*** **Frame Injection Example**

| Register Access | INJ_WR Bits 31:24 | INJ_WR Bits 23:16 | INJ_WR Bits 15:8 | INJ_WR Bits 7:0 |
|---|---|---|---|---|
| INJ_CTRL #1 | GAP_SIZE = 0, ABORT = 0, EOF = 0, SOF = 1, VLD_BYTES = 0 | | | |
| INJ_WR #1 | IH bit 63:56 | IH bit 55:48 | IH bit 47:40 | IH bit 39:32 |
| INJ_WR #2 | IH bit 31:24 | IH bit 23:16 | IH bit 15:8 | IH bit 7:0 |
| INJ_WR #3 | Frame byte 1 (DMAC) | Frame byte 2 (DMAC) | Frame byte 3 (DMAC) | Frame byte 4 (DMAC) |
| INJ_WR #4 | Frame byte 5 (DMAC) | Frame byte 6 (DMAC) | Frame byte 7 (SMAC) | Frame byte 8 (SMAC) |
| … | | | | |
| INJ_CTRL #2 | GAP_SIZE = 0, ABORT = 0, EOF = 1, SOF = 0, VLD_BYTES = 1 | | | |
| INJ_WR #19 | Frame byte 65 (FCS) | Don’t care | Don’t care | Don’t care |
| INJ_CTRL #3 | GAP_SIZE = 0, ABORT = 0, EOF = 0, SOF = 1, VLD_BYTES = 0 | | | |
| INJ_WR #20 | IH bit 63:56 | IH bit 55:48 | IH bit 47:40 | IH bit 39:32 |
| … | | | | |
| INJ_CTRL #4 | GAP_SIZE = 0, ABORT = 0, EOF = 1, SOF = 0, VLD_BYTES = 0 | | | |
| INJ_WR #37 | Frame byte 61 | Frame byte 62 | Frame byte 63 | Frame byte 64 |

### 5.5.3 Frame Interrupts

Software can be interrupted when frame data is available for extraction or when there is room for frames to be injected.

The value of DEVCPU_QS::XTR_DATA_PRESENT.DATA_PRESENT_GRP is provided directly as interrupt inputs to the VCore-III system’s interrupt controller (the XTR_RDY interrupts), so that software can be interrupted when frame data is available for extraction. Using the interrupt controller, these interrupts can be mapped independently to either the VCore-III CPU or external interrupt outputs.

The negated value of DEVCPU_QS::INJ_STATUS.WMARK_REACHED is provided as interrupt inputs to the VCore-III system’s interrupt controller (the INJ_RDY interrupts), so that software can be interrupted when there is room in the IQS. Using the interrupt controller, these can be mapped independently to either the VCore-III CPU or external interrupt outputs.

## 5.6 Frame DMA

The Frame DMA (FDMA) engine is a modified general-purpose DMA engine that extracts and injects frames directly from or to the queue system.

The FDMA has access to the entire VCore-III shared bus. Although DDR2 memory is the most obvious working area for FDMA, transfers can be made across the parallel interface or even the serial interface.

The FDMA engine features eight individual channels that can be configured for either frame injection or frame extraction. A single channel can operate in only one of these modes.

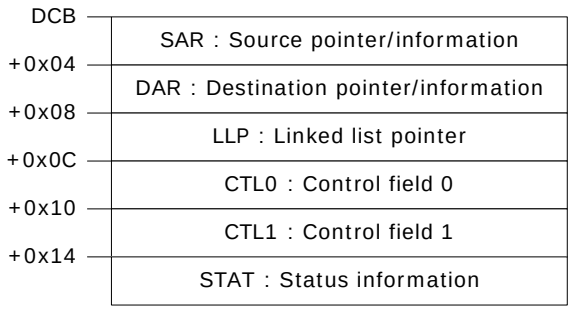
### 5.6.1 DMA Control Block Structures

It is possible to manually instruct the FDMA engine to move arbitrary memory around though register configurations. But most of the time it is desirable to configure transfers though control structures in memory holding information about length, offsets, destinations, pointer to data area, and so forth. The

FDMA engine supports this through the use of DMA Control Block structures (DCB). DCBs are structures that can be linked together to form lists of sequential transfers to be executed by the FDMA.

The following illustration shows the general layout of a DCB.

***Figure 47 •*** **General DCB Layout**



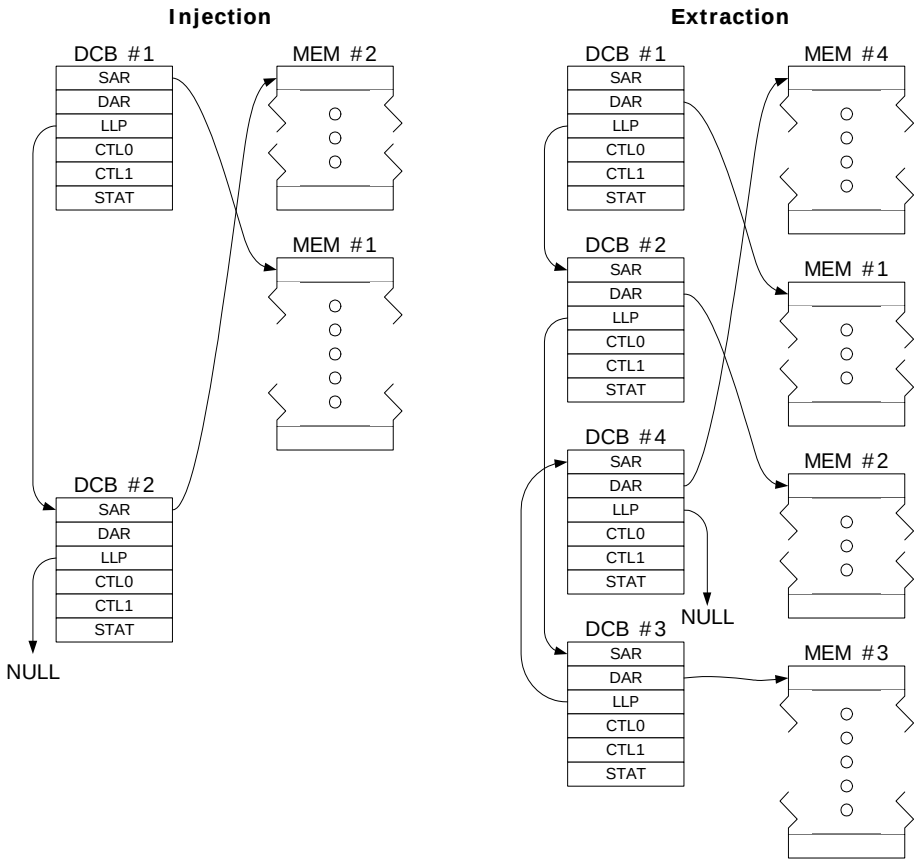

During injection and extraction, one Ethernet frame can be contained within the data area of a single DCB or it can be split across multiple data areas of consecutive DCBs. The data area of one DCB can never contain more than one Ethernet frame.

DCBs and the corresponding data area must be aligned to 32-bit addresses. The encoding of SAR, DAR, CTL0, CTL1, and STAT DCB-fields differs when used for general-purpose data transfers, injection, or extraction. The sections dealing with each transfer type also explain how to configure the DCB fields.

The LLP field in the DCB is always used for linking DCBs into chains. The FDMA engine interprets the LLP field as a 32-bit pointer, when linking DCBs together the LLP of one DCB must point to the SAR field of the next DCB. The last DCB in a chain must always have the LLP field set to NULL (0x00000000).

The following illustration shows two examples of DCB chains with corresponding data areas. It shows how chains of DCBs and corresponding data areas can be placed in any order inside the memory and how the data areas can be of different sizes. For injection, the SAR points to a memory area. For extraction, the DAR points to a memory area.

***Figure 48 •*** **DCB Chain Examples**




The FDMA engine autonomously processes chains of DCBs, so adding DCBs to an active chain requires care. Chains of new DCBs must be constructed separately from the active chain. Channels must be configured for injection or extraction before adding chains of DCBs. For more information about adding lists of DCBs to the FDMA channels or initializing the FDMA engine, see Injection, page 155 and Extraction, page 152.

The switch stores local information about each frame in an internal frame header. When extracting a frame from the queue system, the Extraction Header (EH) is prepended to the frame data (it is located before the actual Ethernet frame). When injecting a frame into the queue system or to an Ethernet port using super priority injection, a suitable Injection Header (IH) must be generated and prepended to the frame data (before the actual Ethernet frame). For information about interpreting and generating the headers for extraction and injection, see CPU Port Module, page 120.

The FDMA engine always interprets frame data in network order (big endian format). This means that no matter which endianess it uses, the CPU must access extracted frame data in network order by doing byte accesses on incrementing byte addresses. The first byte received by the switch is put on the lowest address, and subsequent bytes are put on incrementing byte addresses.

When constructing frames for transmission, the CPU must put frame data on incrementing byte addresses. The first byte that the switch transmits is put on the lowest byte address, and subsequent bytes are put on incrementing byte addresses.

Generally, when the CPU is in big endian mode, it can access fields directly in the frames that are wider than 1 byte. When the CPU is in little endian mode, software must swap bytes when accessing fields wider than 1 byte.

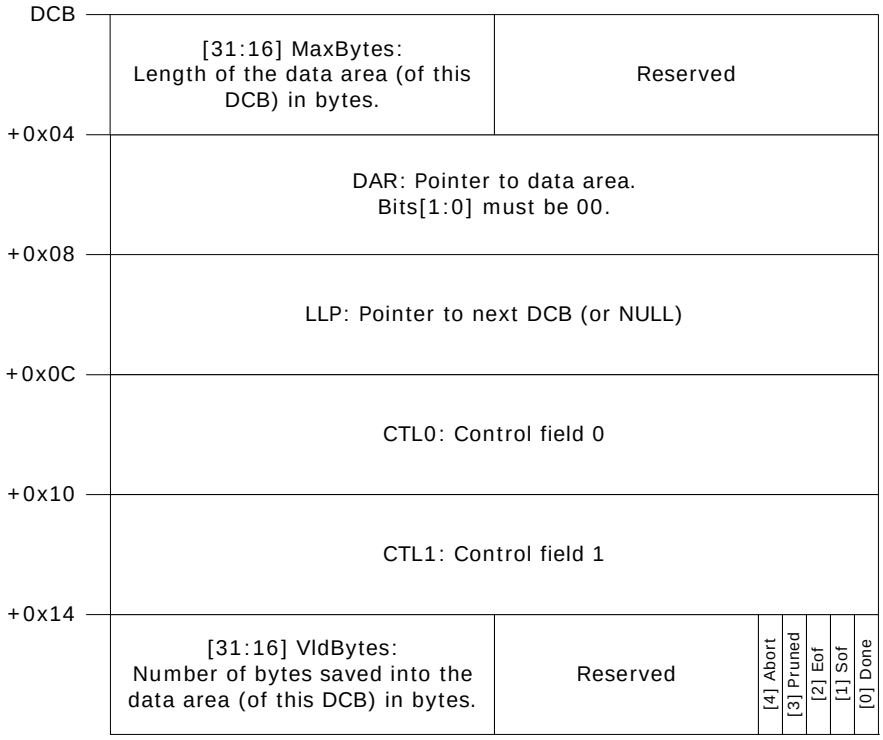
## 5.6.2 Extraction

Frames can be extracted from the queue system. The queue system has eight queues available for extraction. When extracting through the FDMA, the same queues, groups, extraction header, and

mapping apply as when manually extracting through registers. For more information about queues, groups, and frame header information, see Manual Frame Extraction, page 147.

**Note** The “data special values” that are used during manual extraction do not apply when using the FDMA. The FDMA extracts frame data and automatically updates special indications, which are then stored in the DCBs.

***Figure 49 •*** **Extraction DCB Layout**




For extraction, the DAR field holds the pointer to the first 32-bit word in the data area. For more information about LLP, see DMA Control Block Structures, page 150.

The FDMA channels are optimized for bursting data into the working memory. As a result, the minimum data area size for extraction DCBs is 68 bytes (17 32-bit words).

### 5.6.2.1 SAR Field Encoding for Extraction

SAR holds source information and configurations related to extraction of frames. Reserved fields must be set to zero.

The SAR.MaxBytes must be set to the total number of bytes available in the data area of that particular DCB. The value of this field must be divisible by four, that is, bits [1:0] of the field must be 00.

### 5.6.2.2 CTL0 and CTL1 Field Encoding for Extraction

The CTL0 and CTL1 fields are loaded into the corresponding FDMA registers when processing extraction DCBs. Reserved fields must be set to 0.

The least significant bit of CTL0 is a block-interrupt enable field. To achieve optimal performance, use the following values for extraction:

- CTL0: 0x1A40DC24 + (block-interrupt ? 1 : 0)
- CTL1: 0x00000000

When block interrupt is enabled, the FDMA can assert interrupt after a DCB is processed. The interrupt does not stop the FDMA; it can be used by software for detecting arrival of new frames.

#### 5.6.2.3 STAT Field Encoding for Extraction

After a DCB is processed by the FDMA, the STAT field is updated with information about extraction status. When preparing a DCB for extraction, the entire STAT field must be set to 0.

The STAT.Done field is set to 1 after the DCB is processed (this is an indication that the STAT field is valid). STAT.Sof is set if the current DCB contains start-of-frame (when it contains the first byte of the frame header). STAT.Eof is set when the current DCB contains end-of-frame (when it contains the last byte of the frame).

STAT.Pruned is set if the frame was pruned. STAT.Abort is set if the frame was aborted. Frames may be aborted if they are longer than the programmed MTU. For more information about pruning frames, see Manual Frame Extraction, page 147.

The STAT.VldBytes indicates the number of bytes that was saved to the data area of the current DCB.

**Note** When frames are spread across multiple DCBs, the STAT.VldBytes of all the DCBs must be accumulated to get the total frame length.

#### 5.6.2.4 Initialization of FDMA Extraction Channels

There is a one-to-one mapping from extraction groups to FDMA channels (that is, extraction group zero can only be serviced by FDMA channel 0).

Using the extraction queue to group mapping, one FDMA channel can extract from multiple extraction queues. One FDMA channel can handle all extraction queues. For increased performance, use different FDMA channels to separate high-priority and low-priority extraction queues.

Decide on a mapping of extraction queues to FDMA channels. Perform the following steps to enable each FDMA channel (ch) for extraction:

1. Allow QS to control extraction by configuring FDMA:CH[ch]:CFG1.SRC_PER and FDMA:CH[ch]:CFG1.DST_PER to ch. Clear FDMA:CH[ch]:CFG0.HS_SEL_SRC, and set FDMA:CH[ch]:CFG0.HS_SEL_DST.
2. Configure priority through FDMA:CH[ch]:CFG0.CH_PRIOR. The priority controls access to the VCore-III shared bus (the working memory). The FDMA selects between channels with the same priority by using round robin.
3. Configure locking of frame interface by setting FDMA:CH[ch]:CFG0.LOCK_CH and FDMA:CH[ch]:CFG0.LOCK_CH_L to 1.
4. Specify to the frame interface which burst size the FDMA is using by setting ICPU_CFG::FDMA_XTR_CFG[ch].XTR_BURST_SIZE to 2.
5. Allow the FDMA to update the DCBs STAT field by setting FDMA:CH[ch]:CFG1.DS_UPD_EN and FDMA:CH[ch]:DSTATAR to the VCore-III shared bus address of the ICPU_CFG::FDMA_XTR_STAT_LAST_DCB[ch] register.
6. Extraction queues (eq) must be mapped to extraction groups (same as ch). For each extraction queue, configure DEVCPU_QS::XTR_MAP[eq].GRP to ch and set DEVCPU_QS::XTR_MAP[eq].CH_ENA.
7. Enable linked list DCB operation by setting FDMA:CH[ch]:CTL0.LLP_SRC_EN and FDMA_CH[ch]:CTL0.LLP_DST_EN.
8. Configure the FDMA channel for extraction by clearing ICPU_CFG::FDMA_CH_CFG[ch.].USAGE and then setting ICPU_CFG::FDMA_CH_CFG[ch].CH_ENA to enable it.

This procedure assumes that all registers related to the FDMA channel are at their default values before starting configuration. If an extraction channel needs to be reconfigured, reverse all of the above registers to their default values before attempting a new configuration.

#### 5.6.2.5 Extraction of Frames

After initializing an FDMA channel for extraction, frames can be extracted by providing the FDMA with a chain of extraction DCBs. For more information about initializing FDMA channels, see Initialization of FDMA Injection Channels, page 157.

When enabled, the FDMA writes DCBs autonomously, which complicates adding additional DCBs to an enabled FDMA channel (ch). Use the following procedure when adding additional a (null terminated) list of DCBs:

1. Overwrite tail's LLP field (of existing DCB list) with pointer to the head of the new DCB list. Skip this step if there is no existing DCB list for this FDMA channel.
2. Check the state of the FDMA channel. If FDMA::CH_EN_REG.CH_EN[ch]==1, the adding was successful. Do not continue this procedure.
3. If the channel is not enabled, check the STAT field of the head of the new DCB list. If STAT.Done==1, the adding was successful. Do not continue this procedure.
4. If the channel is not enabled and the new DCB list is not used, overwrite FDMA:CH[ch]:LLP with the pointer to the head of the new DCB list. Re-enable the FDMA channel by setting FDMA::CH_EN_REG.CH_EN[ch] and FDMA::CH_EN_REG.CH_EN_WE[ch] at the same time.

**Note** This procedure requires that software keep track of the current DCB list for each FDMA channel. This is part of any software implementation that needs to look at extraction DCBs after they have filled with frame data.

## 5.6.3 Injection

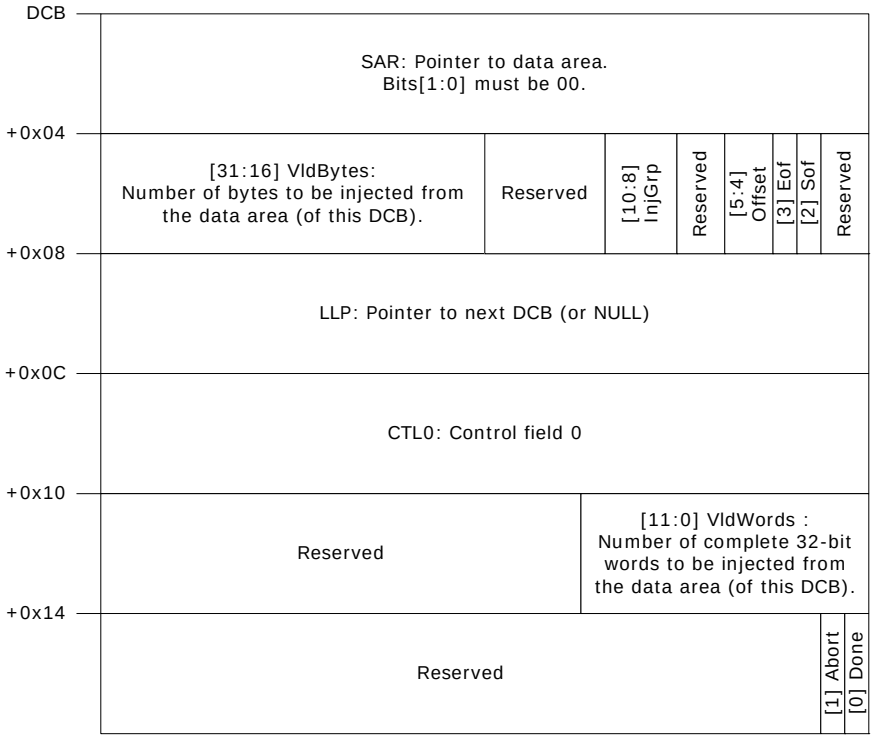
Frames can be injected to the queue system or directly to specific ports. The queue system has two queues (with priorities) available for injection. The same groups and frame header applies when injecting through the FDMA as when manually injecting through registers. For more information about available groups and frame header information, see Manual Frame Injection, page 149.

Each frame has one (and only one) injection group destination encoded directly into the DAR field of the DCB. When a frame is split across multiple DCBs, all the DCBs (for that frame) must be configured with the same destination injection group.

One channel in the FDMA can inject to more than one injection group, however, each injection group must only receive frames from one FDMA channel.

The following illustration shows the detailed layout of an injection DCB. For injection the SAR field holds pointer to the first 32-bit word in the data area. The DAR, CTL0, CTL1, and STAT fields are described in the following sections. For more information about LLP fields, see DMA Control Block Structures, page 150.

***Figure 50 •*** **Injection DCB Layout**



#### 5.6.3.1 DAR Field Encoding for Injection

DAR holds destination information and configurations related to injection of frames. Reserved fields must be set to zero.

The DAR.Sof (start-of-frame) field must be set in each DCB containing the first byte of the frame header. That is; for each frame to be injected, the first DCB for that frame must have DAR.Sof set. The DAR.Eof (end-of-frame) field must be set in each DCB containing the last byte of the frame. That is; for each frame to be injected, the last DCB for that frame must have DAR.Eof set.

**Note** When a frame and its header are contained entirely within a single DCB, that DCB must have both DAR.Sof and DAR.Eof set.

The DAR.Offset field specifies the first valid byte address in the 32-bit word that SAR points to. The following table describes the encoding of the DAR.Offset field.

***Table 119 •*** **DAR.Offset Field Encoding**

| DAR.Offset | Description |
|---|---|
| 0 | Byte address 0 contains the first valid byte. |
| 1 | Byte address 1 contains the first valid byte. |
| 2 | Byte address 2 contains the first valid byte. |
| 3 | Byte address 3 contains the first valid byte. |

The destination group field (DAR.InjGrp) must be set for every DCB.

The DAR.VldBytes field reflects the number of valid bytes in the data area of the DCB. The smallest allowed value is 1, and the largest allowed value is the maximum allowed frame size (that is, MTU) plus the length of the frame header. This means that it is possible to store anything from 1 byte to a complete frame in the data area of one DCB.

DAR.VldBytes does not have to be a multiple of four; the FDMA engine takes care of appropriate buffering and realignment of frame data.

**Important** The DAR.VldBytes field only reflects the number of valid bytes in the data area of the specific DCB. That is, when an Ethernet frame and an internal frame header are contained in multiple DCBs, the DAR.VldBytes field in each individual DCB only indicates the number of valid bytes in the data area of that particular DCB.

#### 5.6.3.2 CTL0 and CTL1 Field Encoding for Injection

The CTL0 and CTL1 fields are loaded into the corresponding FDMA registers when processing injection DCBs. Reserved fields must be set to zero.

The least significant bit of CTL0 is a block-interrupt enable field. CTL1 is set to a ceiling-divide-by-four of VldBytes + Offset. The following values will achieve optimal performance for injection.

- CTL0: 0x1890D924 + (block-interrupt ? 1 : 0)
- CTL1: ((DAR:VldBytes + DAR:Offset + 3) >> 2) & 0x00000FFF

**Note** For injection, block-interrupt is typically enabled for DCBs that contain end-of-frame (where DAR.Eof is set).

#### 5.6.3.3 STAT Field Encoding for Injection

When a DCB is processed by the FDMA, the STAT field is updated with information about injection status. When preparing a DCB for injection, the entire STAT field must be set to 0.

The STAT.Done field is set to 1 when the DCB is processed, which indicates that the STAT field is valid. STAT.Abort is set when injection of the current DCB (or any previous DCBs belonging to the current frame) are aborted by the user (through the ICPU_CFG::FDMA_CFG register).

#### 5.6.3.4 Initialization of FDMA Injection Channels

Any FDMA channel can be configured for frame injection. When an FDMA channel is configured for injection, it can only be used for that purpose. That is, it can no longer be used for extraction or general-purpose transfers.

One FDMA channel can inject to multiple injection groups, however, one injection group must only receive frames from more than one FDMA channel. One FDMA channel can handle all injection groups; however backpressure on any injection group will cause backpressure on the corresponding FDMA channel. For increased performance, separate high-priority and low-priority injection groups by using different FDMA channels.

Decide on a mapping of FDMA channels and injection groups. Perform the following steps to enable each FDMA channel (ch) for injection:

1. Allow QS to control injection by setting FDMA:CH[ch]:CFG1.SRC_PER and FDMA:CH[ch]:CFG1.DST_PER to ch. And setting FDMA:CH[ch]:CFG0.HS_SEL_SRC and FDMA:CH[ch]:CFG0.HS_SEL_DST to zero.
2. Configure priority through FDMA:CH[ch]:CFG0.CH_PRIOR, the priority controls access to the VCore-III shared bus (the working memory). The FDMA selects between channels with the same priority by using round robin.
3. Allow the FDMA to update the DCBs STAT field by setting FDMA:CH[ch]:CFG1.DS_UPD_ENA.
4. Injection groups (ig) which receive frames from the FDMA channel ch must send backpressure to this channel. For each injection group: configure ICPU_CFG::FDMA_INJ_CFG[ig].INJ_GRP_BP_MAP to ch and set ICPU_CFG::FDMA_INJ_CFG[ig].INJ_GRP_BP_ENA.
5. Enable linked list DCB operation by setting FDMA:CH[ch]:CTL0.LLP_SRC_EN and FDMA_CH[ch]:CTL0.LLP_DST_EN.
6. Configure the FDMA channel for injection and then enable it by setting ICPU_CFG::FDMA_CH_CFG[ch].USAGE and ICPU_CFG::FDMA_CH_CFG[ch].CH_ENA.

This procedure assumes that all registers related to the FDMA channel are at their default values before starting configuration. If an injection channel needs reconfiguration, reverse all of the above registers to their default values before attempting a new configuration.

#### 5.6.3.5 Injection of Frames

After initializing an FDMA channel for injection, frames can be injected by providing the FDMA with a chain of injection DCBs. The destination injection group must be specified in the DCB's DAR field. For more information, see Initialization of FDMA Injection Channels, page 157 and DAR Field Encoding for Injection, page 156.

Software must ensure that the FDMA channel only injects to groups that have already been associated with the channel (done during initialization of FDMA injection channels).

When enabled, the FDMA reads DCBs autonomously, which complicates adding additional DCBs to an enabled FDMA channel (ch). Use the following procedure when adding a (null terminated) list DCBs for injection.

1. Overwrite tail's LLP field (of existing DCB list) with pointer to the head of the new DCB list. Skip this step if there is no existing DCB list for this FDMA channel.
2. Check the state of the FDMA channel. If FDMA::CH_EN_REG.CH_EN[ch]==1, the adding was successful. Do not continue this procedure.
3. If channel is not enabled, check the STAT field of the head of the new DCB list. If STAT.Done==1, the adding was successful. Do not continue this procedure.
4. If the channel is not enabled and the new DCB list is not injected, overwrite FDMA:CH[ch]:LLP with the pointer to the head of the new DCB list. Re-enable the FDMA channel by setting FDMA::CH_EN_REG.CH_EN[ch] and FDMA::CH_EN_REG.CH_EN_WE[ch] at the same time.

This procedure requires that software keep track of the current DCB list for each FDMA channel. This should be part of any software implementation that wants to reclaim injection DCBs after they have been injected.

#### 5.6.3.6 Continuous Injection of Frames

The FDMA can be configured for continual injection of frames by linking the tail to the head of a DCB list. This will cause a continuous transmission of all the DCBs in the list. This feature is useful when specific frames are needed for monitoring links between switches in the network, for example, continual transmission of CCM frames.

The following table lists the registers associated with injection frame spacing.

***Table 120 •*** **Injection Frame Spacing Registers**

| Register | Description | Replication |
|---|---|---|
| INJ_FRM_SPC_TMR | Injection frame spacing timer | Per DMA channel |
| INJ_FRM_SPC_TMR_CFG | Reload value for the injection frame spacing timer | Per DMA channel |
| INJ_FRM_SPC_LACK_CNTR | Lack counter | Per DMA channel |
| INJ_FRM_SPC_CFG | Injection frame spacing configuration register | Per DMA channel |

A delay can be inserted between each DCB so that frames are spaced evenly when injected. The delay between the transmissions of DCBs in the list is configured in INJ_FRM_SPC_TMR.TMR. The resulting delay depends on the VCore-III system frequency. The frame space timer is down-counting and the current value of the timer can be read in INJ_FRM_SPC_TMR.TMR.

To enable the frame spacing feature, the INJ_FRM_SPC_CFG.FRM_SPC_ENA must be set. The frame spacing timer can be enabled/disabled using INJ_FRM_SPC_CFG.TMR_ENA.

If the switch queue systems fill-level causes the FDMA transfers to stop for an extended period of time or if the MIPS or DDR controller occupies the AHB bus, the requested frame spacing may not be met. When it is possible to start the transmission again the frames that have been postponed are transmitted without a delay is inserted between them. The number of frames to transmit “unspaced” is counted by the lack counter. The lack counter is incremented every time the frame space timer ticks while frames cannot be transmitted. The lack counter saturates at 511 and cannot go negative, thus up to 511 outstanding frames are supported. The current value of the lack counter can be read in INJ_FRM_SPC_LACK_CNTR.LACK_CNTR.

There should be a one-to-one correspondence between frames and DCBs when configuring the DCB ring. If the frame to be injected spans several DCBs, it will take a frame space timer-tick per DCB to inject the frame.

The frame space timer is 32 bits wide, allowing transmission rates down to 17.1 seconds with a VCore-III system frequency of 250 MHz. If longer transmission rates are required, dummy frames must be inserted in the DCB ring.

## 5.6.4 Frame DMA Interrupt

The Frame DMA generates an interrupt if any of the following events occur:

- When the FDMA tries to access an illegal memory region (this does not occur unless the FDMA was misconfigured). This is an ERR-event.
- When a DCB, with LLP field set to NULL, is processed. This is a TFR-event.
- When a DCB is processed. This is a BLOCK-event.

**Note** Software is most likely interested in getting interrupts when the FDMA finishes processing DCBs. Getting BLOCK events requires enabling of BLOCK interrupt for the (active) extraction channels. The BLOCK-event is useful for reclaiming used injection DCBs or detecting when new frames are extracted from the QS. When interrupt is received, the status of the interrupting channels can be read from FDMA::STATUS_BLOCK. When interrupt has been handled, the event can be cleared by writing to FDMA::CLEAR_BLOCK.

The behavior of BLOCK-events described previously applies directly to ERR and TFR events. Just replace the *_BLOCK registers with _ERR and _TFR, respectively.

## 5.7 External CPU Support

This section describes the handles of the device, which is dedicated to supporting external CPU systems. In addition to the dedicated logic, an external CPU can interact with most of the VCore-III system.

An external CPU attaches to the device through the SI, PI, or MIIM and has access to register targets in the switch core domain. Through these register targets, indirect access into the VCore-III system on the VCore-III SBA is possible. For more information, Access to the VCore-III Shared Bus, page 168. The external CPU can coexist with the internal VCore-III CPU and hardware-semaphores and interrupts are implemented for inter-CPU communication. For more information, see Mailbox and Semaphores, page 169.

### 5.7.1 Register Access and Multimaster Systems

The access time is the time it takes for a CPU interface to read or write a register inside a register target. The access time depends on the target and the number of CPU interfaces that are attempting to access the target. There are two types of targets:

- Fast Register Targets have dedicated logic for each CPU interface, and the interfaces have guaranteed access to the fast targets; the access time is no more than 35 ns.
- Normal Register Targets are accessible by all CPU interfaces. When different interfaces access the same target, each interface competes for access. When a target is accessed by only one CPU interface, the maximum access time is 1.1 µs. When a target is accessed by more than one CPU interface, the access time is increased to no more than 2.2 µs.

Fast Targets are DEVCPU_QS, DEVCPU_ORG, DEVCPU_PI (only accessible through the parallel interface), and the VCore-III registers (ICPU_CFG, UART, and so on). All other register targets in the device are considered Normal Targets.

The VCore-III registers are placed on the VCore-III shared bus and are indirectly accessible to an external CPU through the DEVCPU_GCB register target.

### 5.7.2 Serial Interface in Slave Mode

This section provides information about the function of the serial interface (SI) in slave mode.

The following table lists the registers associated with SI slave mode.

***Table 121 •*** **SI Slave Mode Register**

| Register | Description |
|---|---|
| SI | Configuration of endianess, bit order, and padding |

The serial interface implements a SPI-compatible protocol that allows an external CPU to perform read and write accesses to register targets inside the device. Endianess and bit order is configurable, and several options for high frequencies are supported.

The serial interface is available to an external CPU when the VCore-III CPU does not own the SI. For more information, VCore-III System and CPU Interface, page 131.

The following table lists the pins of the SI interface.

***Table 122 •*** **SI Slave Mode Pins**

| Pin Name | Direction | Description |
|---|---|---|
| SI_nEn | I | Active low chip select |
| SI_Clk | I | Clock input |
| SI_DI | I | Data input (MOSI) |
| SI_DO | O | Data output (MISO) |

SI_DI is sampled on rising edge of SI_Clk. SI_DO is changed on falling edge of SI_Clk. There are no requirements on the logical values of the SI_Clk and SI_DI inputs when SI_nEn is asserted or deasserted, they can be either 0 or 1. SI_DO is only driven during reading when read-data is shifted out of the device.

The external CPU initiates access by asserting chip select and then transmitting one bit read/write indication, one don't care bit, and 22 address bits. For write access, an additional 32 data bits are transmitted. For read access, the external CPU continues to clock the interface while reading out the result.

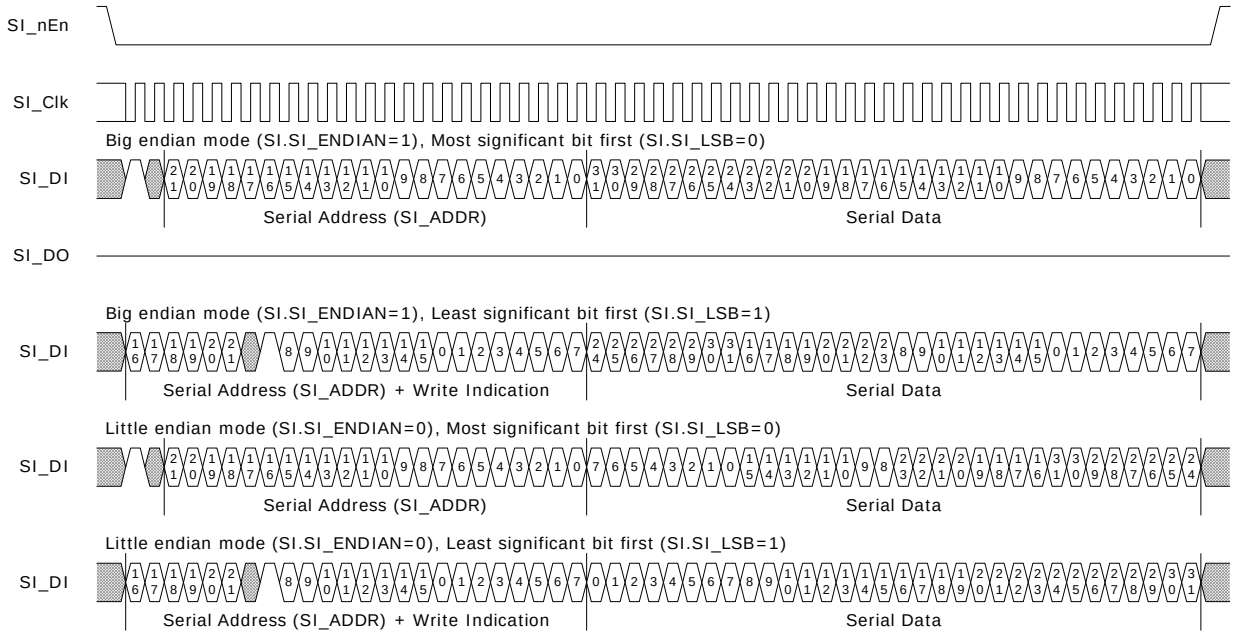
With the register address of a specific register (REG_ADDR), the SI address (SI_ADDR) is calculated:

```
SI_ADDR = (REG_ADDR) – 0×60000000)>>2
```

Data word endianess is configured through SI.SI_ENDIAN. The order of the data bits is configured using SI.SI_LSB. Setting SI.SI_LSB affects both the first 24 bits of the SI command and the 32 bits of data.

The following illustration shows various configurations for write access. The data format during writing, as depicted, is also used when the device is transmitting data during read operations.

***Figure 51 •*** **Write Sequence for SI**




When reading registers using the SI interface, the device needs to prepare read data after receiving the last address bit. The access time of the register that is read must be satisfied before shifting out the first bit of read data. For information about access time, see Register Access and Multimaster Systems, page 159. The external CPU must apply one of the following solutions to satisfy access time:

- Use SI_Clk with a period that is a minimum of twice the access time for the register target. For example, for Normal Targets (single master): 1/(2 × 1.1 µs) = 450 kHz.
- Pause the SI_Clk between shifting of serial address bit 0 and the first data bit with enough time to satisfy the access time for the register target.
- Configure the device to send out enough padding (dummy) bytes before transmitting the read data to satisfy the access time for the register target.

Inserting padding (dummy) bytes is configured in SI.SI_WAIT_STATES. The required number of padding bytes depends on the SI frequency. The SI_DO output is not driven while shifting though padding bytes.

**Note** When using padding bytes, it is usually cumbersome to change the padding configuration on the fly. Then it makes sense to use enough padding to support the worst case access time.

Example: The required number of padding bytes for 20 MHz SI. The clock period at 20 MHz is 50 ns; it will take 50 ns × 8 = 400 ns to shift through one padding byte. For a single master system, the worst-case access time to any register target is 1.1 µs. To satisfy this delay, SI.SI_WAIT_STATES must be configured to at least three. This means that the external CPU must shift a total of 56 bits when reading from the device (the last 32 bits are the read data).

The following illustrations show the options for serial read access. The illustrations show only one mapping of read data, little endian with most significant bit first. Any of the mappings can be configured and apply to read data in the same way as for write data.

***Figure 52 •*** **Read Sequence for SI_Clk Slow**

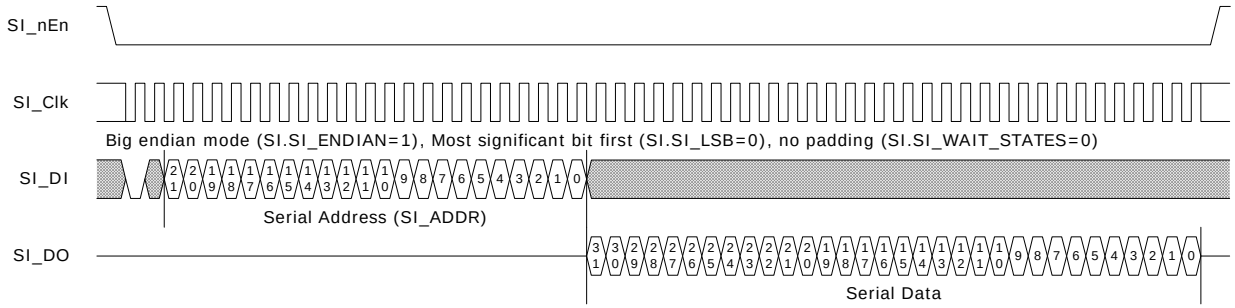



***Figure 53 •*** **Read Sequence for SI_Clk Pause**

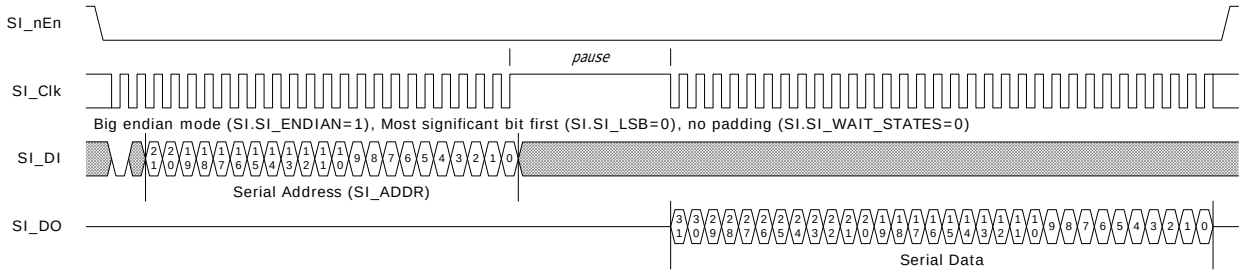
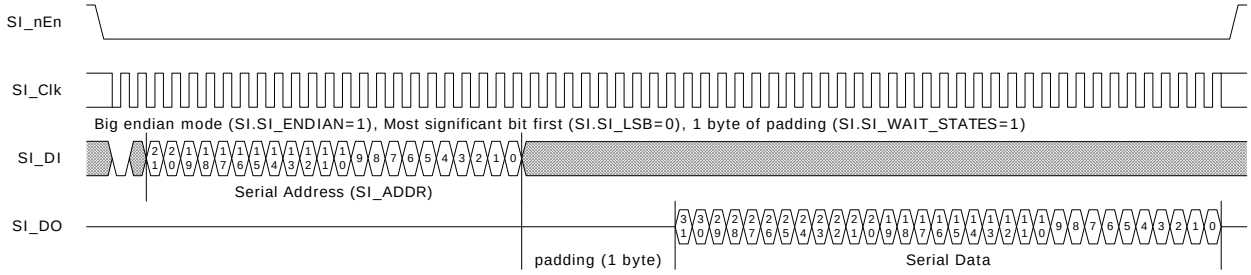



***Figure 54 •*** **Read Sequence for One-Byte Padding**




When using SI, the external CPU must first configure the SI register after power-up, reset, or chip-level soft reset. To configure the device into a known state

1. Write 0 to the SI register.
2. Write the desired configuration using data formatted as little endian with most significant bit first.

## 5.7.3 Parallel Interface in Slave Mode

This section provides information about the functions of the parallel interface (PI) when working in slave mode.

The following table lists the registers associated with PI slave mode.

***Table 123 •*** **PI Slave Mode Registers**

| Register | Description |
|---|---|
| PI_MODE | Controls endianess and done pin polarity |
| PI_CTRL | Configuration of slow access methods |
| PI_CFG | Configuration of PI accesses |
| PI_STAT | Status for PI accesses |
| PI_SLOW_DATA | Slow access registers (two replications) |

The parallel interface allows an external CPU to do read and write access to 32-bit register targets inside the device. Endianess is configurable. Several different access methods are also supported.

All parallel interface pins on the device are overlaid functions on the GPIO interface. PI slave mode is enabled by appropriate configuration of the VCORE_CFG strapping pins. When PI slave mode is enabled, the appropriate GPIO pins are automatically overtaken. For more information, see Overlaid Functions on the GPIOs, page 177. For more information about configuring the VCORE_CFG strapping pins, see VCore-III System and CPU Interface, page 131.

The following table lists the pins of the parallel interface.

***Table 124 •*** **PI Slave Mode Pins**

| Pin Name | I/O | Description |
|---|---|---|
| PI_nCS, GPIO | I | Active low chip select. |
| PI_Addr[3:0], GPIO | I | These are the address lines, PI_Addr[1:0] can be left unconnected unless auto (sub-word) addressing is disabled. |
| PI_nWR, GPIO | I | Active low write enable. |
| PI_nOE, GPIO | I | Active low output enable. |
| PI_Data[7:0], GPIO | I/O | These are the data lines. |
| PI_nDone, GPIO | OZ | An external device can use this output to detect when transfers are done, and thereby optimize the speed of transfers. |

PI_Data is driven by the device when PI_nCS and PI_nOE are both asserted. PI_nDone is driven when PI_nCS is asserted. The drive of PI_nDone is extended a short period after PI_nCS is deasserted, which gives the device time to “park” the PI_nDone signal as inactive before it is released.

The external CPU initiates access by asserting chip select and then driving the appropriate control signals. The timing of the parallel interface is asynchronous; it takes the device from 5 ns to 15 ns to detect an asserted chip select. After detecting chip select, the device waits a configurable amount of time (PI_CFG.PI_WAIT) and then sample PI_Addr, PI_nWR, and PI_Data (PI_Data is only sampled when writing to the device).

To access registers in the device, 32-bit reads and writes must be performed. Because the PI width is 8 bits, four sequential PI accesses are needed for each register access. By default, the parallel interface automatically keeps track of outstanding accesses and aligns current PI_Data appropriately. This feature is called auto (subword) addressing, which is when active PI_Addr[1:0] pins are don’t care and can be left unconnected. Automatic (sub-word) addresses can be disabled by setting PI_MODE.ADDR_AUTO_DIS. When disabled, the external CPU must drive PI_Addr[1:0].

With the register address of a specific register (REG_ADDR), the PI address (PI_ADDR) is calculated as:

```
PI_ADDR = REG_ADDR - 0x60000000
```

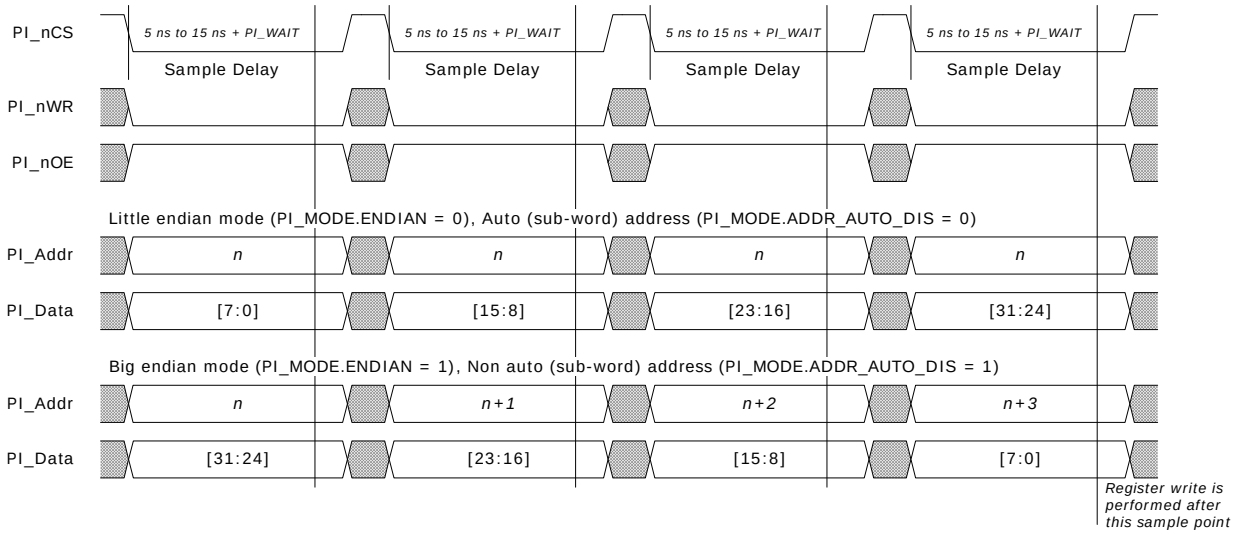
**Note** The parallel interface is byte addressable, because 8-bit mode is supported. However, by default, PI_Addr[1:0] is not used due to the auto (subword) address feature.

The devices only have the lower four address bits mapped to GPIO pins. A windowed mode is used for accessing the full range of parallel addresses (PI_ADDR). For more information, see Windowed Addressing Mode, page 165.

The endianess of the parallel interface is configured through PI_MODE.ENDIAN. The following two illustrations show two configurations of the parallel interface, and how, when auto (subword) addressing is enabled, PI_Addr[1:0] is a don't care and to be left unconnected (this is why the first configuration uses the same address for all accesses). For second configuration the external CPU drives PI_Addr[1:0] (and increments these for each 8-bit write access).

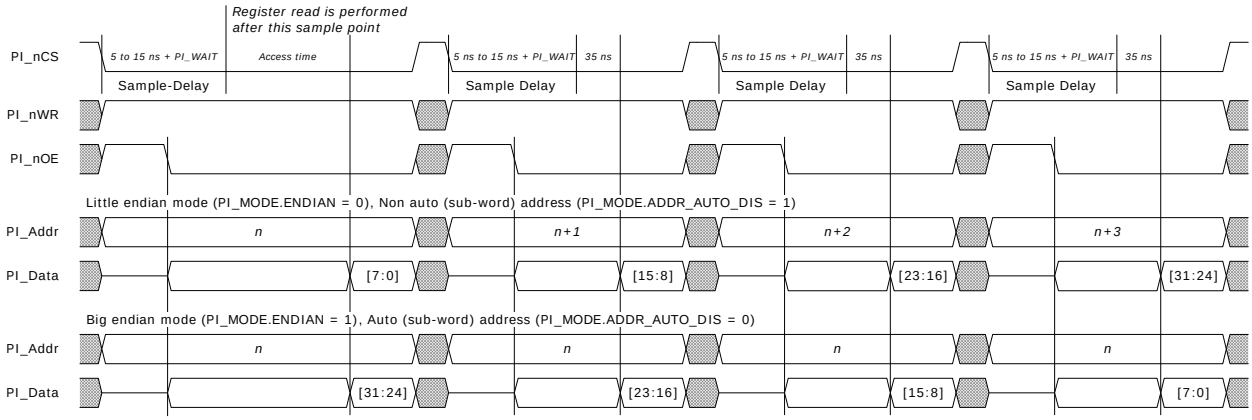
The following illustration shows the write sequence for the parallel interface. This example depicts that the actual register write is performed after the last sample point, which means that a subsequent access on the parallel interface must not be performed until the access is done. For more information about access time for different register targets, see Register Access and Multimaster Systems, page 159.

***Figure 55 •*** **Write Sequence for PI**




When reading registers using the parallel interface, the first access on the parallel interface is subjected to fetching of register data. The access time of the register, which is read, must be satisfied before the external CPU can sample the read-data. The remaining accesses (reading the rest of the 32-bit register data) have an access time equal to reading from the DEVCPU_PI target. For more information about access time see Register Access and Multimaster Systems, page 159.

***Figure 56 •*** **Read Sequence for PI**




When using PI, the first thing the external CPU must do after power-up, reset, or chip-level soft reset is to configure the PI_MODE register. Perform two writes to PI_MODE register with the desired configuration mirrored throughout the entire 32-bit data word. For more information, see the PI_MODE register. information.

### 5.7.3.1 Using PI_nDone to Speed Up Register Access

The parallel interface provides the PI_nDone signal, which is driven during all accesses on the parallel interface.

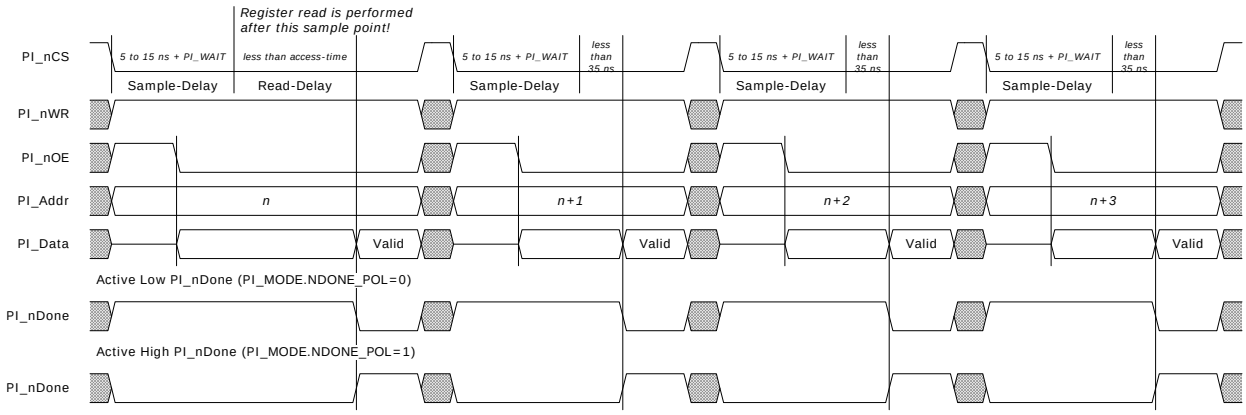
The PI_nDone signal shows when the parallel interface is done with a given access. By monitoring the PI_nDone signal, and terminating accesses when the PI_nDone signal is asserted, an external CPU uses exactly the amount of time that each access requires. The polarity of the PI_nDone is configurable through PI_MODE.NDONE_POL.

When using PI_nDone, an external CPU does not have to take any precautions with regards to the access time parameter. For more information, see Register Access and Multimaster Systems, page 159.

**Note** The access time is a worst-case parameter. Access to Normal Targets when using PI_nDone is typically done after 0.5 µs. Using PI_nDone significantly speeds up access to the parallel interface.

The following illustration shows an example of reading with the PI_nDone signal.

***Figure 57 •*** **PI Read Sequence Using PI_nDone**




Writing is similar, however, because the parallel interface cache writes, the actual register write occurs after the last write access to a given register. When using PI_nDone timing, the subsequent access after writing to the device can be performed immediately. The PI_nDone signaling takes in account additional delay required for finishing the (previous) write access.

#### 5.7.3.2 Using Paged Access to Get Fixed PI Timing

By enabling paged access, all parallel access to the devices have timing as if they were directly accessing the DEVCPU_PI target (Fast Target). This means that an external CPU does not have to change I/O timing, depending on the register target that is accessed.

Paging is enabled using PI_CTRL.SLOW_ENA. Access to any register target other than DEVCPU_PI is paged. Paging works by storing read and write values internally inside the parallel interface, write values are cached, and read values, when ready, are available in the PI_SLOW_DATA registers. Which of the PI_SLOW_DATA registers to use for a specific paged access is configured in PI_CTRL.SLOW_IDX.

The external CPU can see when accesses are done by polling PI_STAT.SLOW_BUSY field corresponding to the PI_CTRL.SLOW_IDX is used. The PI_STAT.SLOW_DONE field shows when read data is available in the corresponding PI_SLOW_DATA register. The PI_STAT.SLOW_DONE indications are also available to the VCore-III interrupt controller through the PI_SD0 and PI_SD1 interrupts. By means of the interrupt controller, done-indications can be mapped to external interrupt outputs so that an external CPU can use these when waiting for paged reads to complete.

**Note** The PI_SLOW_DATA, PI_STAT.SLOW_BUSY, and PI_STAT.SLOW_DONE are replicated two times, which allows two different threads on an external CPU to use their own dedicated paging logic. This is useful when, for example, an interrupt thread needs access to the device in parallel with normal device access. The interrupt routine must configure PI_CTRL.SLOW_IDX at the start of the interrupt-routine and reset it before returning.

Paged accesses are cached and handled internally inside the parallel interface. Use the following sequence to perform a paged read of the DEVCPU_GCB::GENERAL_PURPOSE register:

1. Perform a register read from DEVCPU_GCB::GENERAL_PURPOSE, ignore the read-data.
2. Wait until the read access is done. Either poll PI_STAT.SLOW_DONE or examine external interrupt output.
3. Read the result of the read from the PI_SLOW_DATA register corresponding to the PI_CTRL.SLOW_IDX that was set when the register read was performed.

Writing is similar to reading; again the same register is used as an example:

1. Perform register write to DEVCPU_GCB::GENERAL_PURPOSE.
2. Do not start a new access until the write access is done, poll PI_STAT.SLOW_BUSY until done.

When mapping done indications using the VCore-III interrupt controller, it is recommended that you disable interrupt stickiness so that reading the PI_SLOW_DATA registers also clears the external interrupt indication. For more information, see Interrupt Controller, page 184.

#### 5.7.3.3 Windowed Addressing Mode

The parallel interface allows configuration of address offset through an address window. The address window is accessed by writing to or reading from the highest register address (highest possible 32-bit word address). When windowed addressing is used; the address window must be configured prior to accessing a device register. The address window is not changed by hardware; subsequent accesses to the same register do not require re-configuration of the address window.

**Note** The internal register address is 22 bits wide (excluding the byte addresses). Only the lowest four parallel address pins are provided on the GPIO interface. All other addresses are tied high internally in the parallel interface. When an external CPU drives both PI_Addr[3:2] pins high, it is accessing the address window register.

The address window register is physically a part of the parallel interface and is not listed in the register list.

An external CPU that cannot or does not want to drive all PI_Addr wires can use widowed mode to access the device. Unused PI_Addr connections must be left floating or tied high.

By using both the auto (sub word) addressing feature and address window mode; an external CPU can connect to as few as one address pin (PI_Addr[2]) and still control the device.

The address window register is all-ones per default. If bits [23:3] in the address window register are set to 0, then the corresponding parallel address [23:3] are also forced to 0. If bit [2] in the address window

register set to 0, then parallel address [2] is forced to 1. Bits [31:26] and [1:0] are not implemented and read as zeros; bits [25:24] must always be written to 11.

Example: Read from to DEVCPU_ORG::ERR_CNTS using PI_Addr[3:2]. All other PI_Addr pins have been left floating and auto (sub word) addressing has not been disabled. DEVCPU_ORG has id 0 and ERR_CNTS has register address 3. After programming address window to 0x03000008 (by writing to PI_Addr[3:2] = 11),ERR_CNTS is accessible on PI_Addr[3:2] = 01.

## 5.7.4 MIIM Interface in Slave Mode

This section provides the functional aspects of the MIIM slave interface.

**Note:** The MIIM slave I/F, due to its low bandwidth, is not aimed at supporting or recommended for managed switch applications.

The MIIM slave interface allows an external CPU to perform read and write access to the register targets inside the device. Register access is done indirectly, because the address and data fields of the MIIM protocol is less than those used by the register targets. Transfers on the MIIM interface are using the Management Frame Format protocol specified in IEEE 802.3, Clause 22.

The MIIM slave pins on the device are overlaid functions on the GPIO interface. MIIM slave mode is enabled by configuring the appropriate VCORE_CFG strapping pins. For more information, see VCore-III System and CPU Interface, page 131. When MIIM slave mode is enabled, the appropriate GPIO pins are automatically overtaken. For more information, see Overlaid Functions on the GPIOs, page 177.

The following table lists the pins of the MIIM slave interface.

***Table 125 •*** **MIIM Slave Pins**

| Pin Name | I/O | Description |
|---|---|---|
| MDC_SLV, GPIO | I | MIIM slave clock input |
| MDIO_SLV, GPIO | I/O | MIIM slave data input/output |
| MIIM_SLV_ADDR, GPIO | I | MIIM slave address select |

MDIO_SLV is sampled or changed on the rising edge of MDC_SLV by the MIIM slave interface.

The MIIM slave can be configured to answer on two different PHY addresses using the MIIM_SLV_ADDR pin. Setting the MIIM_SLV_ADDR pin to 0 configures the MIIM slave to use PHY address 0, and setting it to 1 configures the MIIM slave to use PHY address 31.

The MIIM slave has seven 16-bit MIIM registers defined as listed in the following table.

***Table 126 •*** **MIIM Registers**

| Register Address | Register Name | Description |
|---|---|---|
| 0 | ADDR_REG0 | Bit 15:0 of the address to read or write. The address field must be formatted as a word address. |
| 1 | ADDR_REG1 | Bit 31:16 of the address to read or write. |
| 2 | DATA_REG0 | Bit 15:0 of the data to read or write. Returns 0x0000 if a register read error occurred. |
| 3 | DATA_REG1 | Bit 31:16 of the data to read or write. The read or write operation is initiated after this register is read or written. Returns 0x8000 if read while busy or a register read error occurred. |
| 4 | DATA_REG1_INCR | Bit 31:16 of data to read or write. The read or write operation is initiated after this register is read or written. When the operation is complete, the address register is incremented by one. Returns 0x8000 if read while busy or if a register read error occurred. |

*Table 126 •* **MIIM Registers *(continued)***

| Register Address | Register Name | Description |
|---|---|---|
| 5 | DATA_REG1_INERT | Bit 31:16 of data to read or write. Reading or writing to this register will not cause a register access to be initiated. Returns 0x8000 if a register read error occurred. |
| 6 | STAT_REG | The status register gives the status of any ongoing operations.<br>Bit 0: Busy - Is set while a register read/write operation is in progress.<br>Bit 1: Busy_rd - the busy status during the last read or write operation.<br>Bit 2: Err - Is set if a register access error occurred.<br>Others: Reserved. |

A 32-bit register read or write transaction over the MIIM interface is done indirectly due to the limited data width of the MIIM frame. First, the address of the register inside the device must be set in the two 16-bit address registers of the MIIM slave using two MIIM write transactions. Afterwards the two 16-bit data registers can be read/written to access the data value of the register inside the device. Thus, it requires up to four MIIM transactions to perform a single read or write operation on a register target.

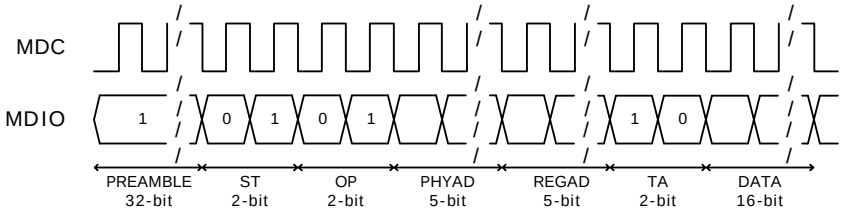
The address of the register to read/write is set in registers ADDR_REG0 and ADDR_REG1. The data to write to the register pointed to by the address in ADDR_REG0 and addr_reg1 is first written to DATA_REG0 and then to DATA_REG1. When the write transaction to DATA_REG1 is completed, the MIIM slave initiates the register transaction.

With the register address of a specific register (REG_ADDR), the MIIM address (MIIM_ADDR) is calculated as:

```
MIIM_ADDR = (REG_ADDR - 0x60000000)>>2
```

The following illustration shows a single MIIM write transaction on the MIIM interface.

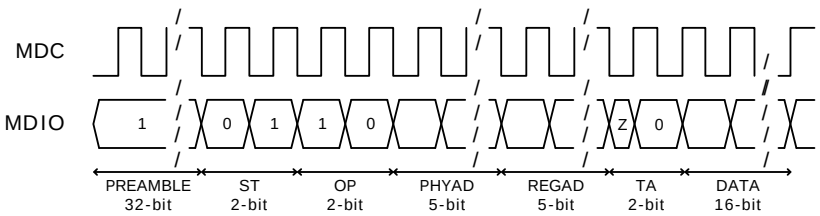
*Figure 58 •* **MIIM Slave Write Sequence**




A reading transaction is done in a similar way. First, read the DATA_REG0 and then read the DATA_REG1. As with a write operation. The register transaction is not initiated before the DATA_REG1 register is read. In other words, the returned read value is from the previous read transaction.

The following illustration shows a single MIIM read transaction on the MIIM interface.

*Figure 59 •* **MIIM Slave Read Sequence**



## 5.7.5 Access to the VCore-III Shared Bus

This section provides information about how to access the VCore-III shared bus (SBA) from an external CPU. The following table lists the registers associated with the VCore-III shared bus access.

***Table 127 •*** **VCore-III Shared Bus Access Registers**

| Register | Description |
|---|---|
| VA_CTRL | Status for ongoing accesses |
| VA_ADDR | Configuration of shared bus address |
| VA_DATA | Data register |
| VA_DATA_INCR | Data register, access increments VA_ADDR |
| VA_DATA_INERT | Data register, access does not start new accesses |

An external CPU perform 32-bit reads and writes to the SBA through the VCore Access (VA) registers. In the VCore-III system, there is a dedicated master on the shared bus that handles VA accesses. For information about arbitration between masters on the shared bus, see Shared Bus Arbitration, page 135.

The SBA address is configured in VA_ADDR. Accessing the VA_DATA register starts an SBA access. Writing to VA_DATA starts a write with the 32-bit value that was written to VA_DATA. Reading from VA_DATA returns the current value of the register and starts a read access, when the read-access completes the result will automatically be stored in the VA_DATA register.

The VA_DATA_INCR register behaves like VA_DATA, except that after starting an access the VA_ADDR register is incremented by 4 (so that it points to the next word address in the SBA domain). Reading from the VA_DATA_INCR register returns the value of VA_DATA, writing to VA_DATA_INCR overwrites the value of VA_DATA.

Note By using VA_DATA_INCR, sequential addresses can be accessed without having to manually increment the VA_ADDR register between each access.

The VA_DATA_INERT register provides direct access to the VA_DATA value without starting accesses on the SBA. Reading from the VA_DATA_INERT register returns the value of VA_DATA, writing to VA_DATA_INERT overwrites the value of VA_DATA.

The VCore-III shared bus is capable of returning error-indication when illegal register regions are accessed. If a VA access result in an error-indication from the SBA, the VA_CTRL.VA_ERR field is set, and the VA_DATA is set to 0x80000000.

**Note** SBA error indications only occur when non-existing memory regions or illegal registers are accessed. It does not occur during normal operation, so the VA_CTRL.VA_ERR indication is useful during debugging only.

Example: Reading from ICPU_CFG::GRP[1] through the VA registers. The ICPU_GPR register is the second register in the SBA VCore-III Registers region. Set VA_ADDR to 0x70000004, read once from VA_DATA (and discard the read-value). Wait until VA_CTRL.VA_BUSY is cleared, then VA_DATA contains the value of the ICPU_CFG::GRP[1] register. Using VA_DATA_INERT (instead of VA_DATA) to read the data is appropriate because this does not start a new SBA access.

### 5.7.5.1 Optimized Reading

SBA access is typically much faster than the CPU interface, which is used to access the VA registers. The VA_DATA register (VA_DATA_INCR and VA_DATA_INERT) return 0x80000000 while VA_CTRL.VA_BUSY is set. This means that it is possible to skip checking for busy between read access to SBA.

For example, after initiating a read access from SBA, software can proceed directly to reading from VA_DATA, VA_DATA_INCR, or VA_DATA_INERT.

- If the second read is different from 0x80000000; then the second read returned valid read data (the SBA access was done before the second read was performed).

- If the second read is equal to 0x80000000; VA_CTRL must be read.

  If VA_CTRL.VA_BUSY_RD is cleared (and VA_CTRL.VA_ERR_RD is also cleared), then 0x80000000 is the actual read data

  If VA_CTRL.VA_BUSY_RD is set, the SBA access was not yet done at the time of the second read. Start over again by repeating the read from VA_DATA.

Optimized reading can be used for single-read access (reading VA_DATA and then VA_DATA_INERT). For sequential reads (reading VA_DATA_INCR several times), the VA_ADDR is only incremented on successful (non-busy) reads.

## 5.7.6 Mailbox and Semaphores

This section provides information about the semaphores and mailbox features for CPU to CPU communication. The following table lists the registers associated with mailbox and semaphore.

***Table 128 •*** **Mailbox and Semaphore Registers**

| Register | Description |
|---|---|
| SEMA | Taking of semaphores, replicated per semaphore. |
| SEMA_FREE | Current status for all semaphores. |
| SEMA_INTR_ENA | Enable software interrupt on free semaphores. |
| SEMA_INTR_ENA_CLR | Atomic clear of the SEMA_INTR_ENA register. |
| SEMA_INTR_ENA_SET | Atomic set of the SEMA_INTR_ENA register. |
| SW_INTR | Asserting of software interrupts. |
| MAILBOX | Mailbox. |
| MAILBOX_CLR | Atomic clear of bits in the mailbox register. |
| MAILBOX_SET | Atomic set of bits in the mailbox register. |

The devices implements eight independent semaphores. The semaphores are controlled through the SEMA register. The SEMA register is replicated once per semaphore; SEMA[0] corresponds to the first semaphore, SEMA[1] the second semaphore, and so on.

Any CPU can attempt to take a semaphore n by reading SEMA[n].SEMA. If the result is 1, the semaphore was successfully taken and is now owned by the CPU. If the result is 0, the semaphore was not free. After a CPU successfully takes a semaphore, all additional reads from the corresponding SEMA register will return 0. To release semaphore *n*, a CPU must write 1 to SEMA[n].SEMA.

**Note** Any CPU can release semaphores; it does not have to the one that has taken the semaphore, this allows implementation of handshaking protocols.

The current status for all semaphores is available in SEMA_FREE.SEMA_FREE.

A software interrupt can be generated when one or more semaphores are free. Interrupt is enabled in SEMA_INTR_ENA.SEMA_INTR_ENA, atomic set and clear are possible through SEMA_INTR_ENA_CLR and SEMA_INTR_ENA_SET. Semaphores [3:0] can trigger SW0 interrupt when enabled and semaphores [7:4] can trigger SW1 interrupt.

The currently interrupting semaphores are available through SEMA_INTR_ENA.SEMA_INTR_IDENT; this field is the result of a logical AND between SEMA_INTR_ENA.SEMA_INTR_ENA and SEMA_FREE.SEMA_FREE.

In addition to interrupting on free semaphores, a software interrupt can be manually set by writing to SW_INTR.SW0_INTR or SW_INTR.SW1_INTR, these fields are self-clearing.

**Note** Software interrupts (SW0 and SW1) can be mapped independently by means of the VCore-III interrupt controller to either VCore-III CPU or external interrupt outputs.

The mailbox is a 32-bit register that can be set and cleared atomically using any CPU interface (including the VCore-III CPU). The MAILBOX register allows reading (and writing) of the current mailbox value. Atomic clear of specific bits in the mailbox register is done by writing a mask to MAILBOX_CLR. Atomic setting of specific bits in the mailbox register is done by writing a mask to MAILBOX_SET.

# 5.8 VCore-III System Peripherals

This section describes the subblocks of the VCore-III system. They are primarily intended to be used by the VCore-III CPU. However, an external CPU can access and control these through the shared bus.

## 5.8.1 Timers

This section provides information about the timers. The following table lists the registers associated with timers.

***Table 129 •*** **Timer Registers**

| Register | Description | Replication |
|---|---|---|
| TIMER_CTRL | Enable/disable timer | Per timer |
| TIMER_VALUE | Current timer value | Per timer |
| TIMER_RELOAD_VALUE | Value to load when wrapping | Per timer |
| TIMER_TICK_DIV | Common timer-tick divider | None |

There are three decrementing 32-bit timers in the VCore-III system that run from a common divider. The common divider is driven by a fixed 250 MHz clock and can generate timer ticks in the range of 0.1 µs (10 MHz) to 1 ms (1 kHz), configurable through TIMER_TICK_DIV. The default timer tick is 100 µs (10 kHz).

**Note** The timers are independent of the VCore-III CPU frequency, because the common divider uses a fixed clock.

Software can access each timer value through the TIMER_VALUE registers. These can be read or written at any time, even when the timers are active.

When a timer is enabled through TIMER_CTRL.TIMER_ENA, it decrements from the current value until it reaches zero. An attempt to decrement a TIMER_VALUE of zero generates interrupt and assigns TIMER_VALUE to the contents of TIMER_RELOAD_VALUE. Interrupts generated by the timers are send to the VCore-III interrupt controller. From here, interrupts can be forwarded to the VCore-III CPU or to an external CPU. For more information, see Interrupt Controller, page 184.

By setting TIMER_CTRL.ONE_SHOT_ENA the timer disables itself after generating one interrupt. When this field is cleared, timers will decrement, interrupt, and reload indefinitely (or until disabled by software, that is, by clearing of TIMER_CTRL.TIMER_ENA).

A timer can be reloaded from TIMER_RELOAD_VALUE at the same time as it is enabled by setting both TIMER_CTRL.FORCE_RELOAD and TIMER_CTRL.TIMER_ENA.

Example: Configure Timer0 So That It Interrupts Every 1 ms. With the default timer tick of 100 µs ten timer ticks are needed for a timer that wraps every 1 ms. Configure TIMER_RELOAD_VALUE[0] to 0x9. Then enable the timer and force a reload by setting TIMER_CTRL[0].TIMER_ENA and TIMER_CTRL[0].FORCE_RELOAD at the same time.

## 5.8.2 UART

This section provides information about the UART (Universal Asynchronous Receiver/Transmitter) controller.

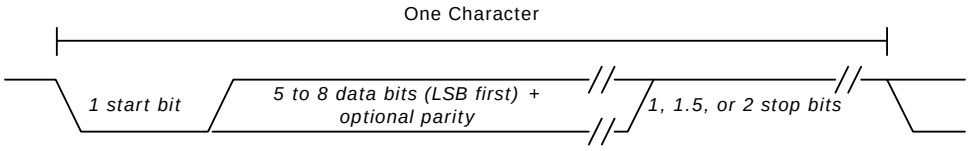
The following table lists the registers associated with the UART.

***Table 130 •*** **UART Registers**

| Register | Description |
|---|---|
| RBR_THR | Receive buffer/transmit buffer/Divisor (low) |
| IER | Interrupt enable/Divisor (high) |
| IIR_FCR | Interrupt identification/FIFO control |
| LCR | Line control |
| MCR | Modem control |
| LSR | Line status |
| MSR | Modem status |
| SCR | Scratchpad |
| USR | UART status |

The VCore-III system UART is functionally based on the industry-standard 16550 UART (RS232 protocol). This implementation features a 16-byte receive and a 16-byte transmit FIFO.

***Figure 60 •*** **UART Timing**




The number of data-bits, parity, parity-polarity, and stop-bit length are all programmable using LCR.

The UART pins on the devices are overlaid functions on the GPIO interface. Before enabling the UART, the VCore-III CPU must enable overlaid modes for the appropriate GPIO pins. For more information, see For more information about enabling the overlaid functionality of the GPIOs, see Overlaid Functions on the GPIOs, page 177.

The following table lists the pins of the UART interface.

***Table 131 •*** **UART Interface Pins**

| Pin Name | I/O | Description |
|---|---|---|
| UART_RX/ GPIO_31 | I | UART receive data |
| UART_TX/GPIO_30 | O | UART transmit data |

The baud rate of the UART is derived from the VCore-III system frequency. The divider value is indirectly set through the RBR_THR and IER registers. The baud rate is equal to the VCore-III system clock frequency divided by sixteen multiplied by the value of the baud rate divisor. A divider of zero disables the baud rate generator and no serial communications occur. The default value for the divisor register is zero.

Example: Configure a baud rate of 9600 in a 125 MHz system. To generate a baud rate of 9600, the divisor register must be set to 0x32E (125  MHz/(16 × 9600 Hz)). Set LCR.DLAB and write 0x2E to RBR_THR and 0x03 to IER (this assumes that the UART is not in use). Finally, clear LCR.DLAB to change the RBR_THR and IER registers back to the normal mode.

By default, the FIFO mode of the UART is disabled. Enabling the 16-byte receive and 16-byte transmit FIFOs (through IIR_FCR) is recommended.

**Note** Although the UART itself supports RTS and CTS, these signals are not available on the pins of the device.

#### 5.8.2.1 UART Interrupt

The UART can generate interrupt whenever any of the following prioritized events are enabled (through IER):

- Receiver error
- Receiver data available
- Character timeout (in FIFO mode only)
- Transmit FIFO empty or at or below threshold (in programmable THRE interrupt mode)

When an interrupt occurs, the IIR_FCR register can be accessed to determine the source of the interrupt. Note that the IIR_FCR register has different purposes when reading or writing. When reading, the interrupt status is available in bits 0 through 3. For more information about interrupts and how to handle them, see the IIR_FCR register description.

Example: Enable Interrupt When Transmit FIFO is Below One-Quarter Full. To get this type of interrupt, the THRE interrupt must be used. First, configure TX FIFO interrupt level to one-quarter full by setting IIR_FCR.TET to 10; at the same time, ensure that the IIR_FCR.FIFOE field is also set. Set IER.PTIME to enable the THRE interrupt in the UART. In addition, the VCore-III interrupt controller must be configured for the CPU to be interrupted. For more information, see Interrupt Controller, page 184.

### 5.8.3 Two-Wire Serial Interface

This section provides information about the functions of the two-wire serial interface controller.

The following table lists the registers associated with the two-wire serial interface.

***Table 132 •*** **Two-Wire Serial Interface Registers**

| Register | Description |
|---|---|
| CFG | General configuration |
| TAR | Target address |
| SAR | Slave address |
| DATA_CMD | Receive/transmit buffer and command |
| SS_SCL_HCNT | Standard speed high time clock divider |
| SS_SCL_LCNT | Standard speed low time clock divider |
| FS_SCL_HCNT | Fast speed high time clock divider |
| FS_SCL_LCNT | Fast speed low time clock divider |
| INTR_STAT | Masked interrupt status |
| INTR_MASK | Interrupt mask register |
| RAW_INTR_STAT | Unmasked interrupt status |
| RX_TL | Receive FIFO threshold for RX_FULL interrupt |
| TX_TL | Transmit FIFO threshold for TX_EMPTY interrupt |
| CLR_* | Individual CLR_* registers are used for clearing specific interrupts. See register descriptions for corresponding interrupt. |
| CTRL | Control register |
| STAT | Status register |
| TXFLR | Current transmit FIFO level |
| RXFLR | Current receive FIFO level |
| TX_ABRT_SOURCE | Arbitration sources |
| SDA_SETUP | Data delay clock divider |

*Table 132 •* **Two-Wire Serial Interface Registers *(continued)***

| Register | Description |
|---|---|
| ACK_GEN_CALL | Acknowledge of general call |
| ENABLE_STATUS | General two-wire serial controller status |
| TWI_CONFIG | Configuration of SDA hold-delay |

The two-wire serial interface controller is compatible with the industry standard two-wire serial interface protocol. The controller supports standard speed up to 100 kbps and fast speed up to 400 kbps. Multiple bus masters, as well as both 7-bit and 10-bit addressing are also supported.

By default, the two-wire serial interface controller operates as master only (CFG.MASTER_ENA), however, slave mode can be enabled (CFG.SLAVE_DIS). In slave mode, the controller generates an interrupt when addressed by an external master. For read requests, the controller then halts the two-wire serial bus until the VCore-III CPU has processed the request and provided a response (reply-data) to the controller. The slave addresses (SAR) of the two-wire serial interface controller must be unique on the two-wire serial interface bus. This must be configured before enabling slave mode. For information about addresses that have a special meaning on the bus, see Two-Wire Serial Interface Addressing, page 174.

The two-wire serial interface pins on the devices are overlaid functions on the GPIO interface. Before enabling the two-wire serial interface, the VCore-III CPU must enable overlaid functions for the appropriate GPIO pins. For more information, see Overlaid Functions on the GPIOs, page 177.

The following table lists the pins of the two-wire serial interface.

*Table 133 •* **Two-Wire Serial Interface Pins**

| Pin Name | I/O | Description |
|---|---|---|
| TWI_SCL, GPIO | O | Two-wire serial interface clock, open-collector output. |
| TWI_SDA, GPIO | I/O | Two-wire serial interface data, open-collector output. |

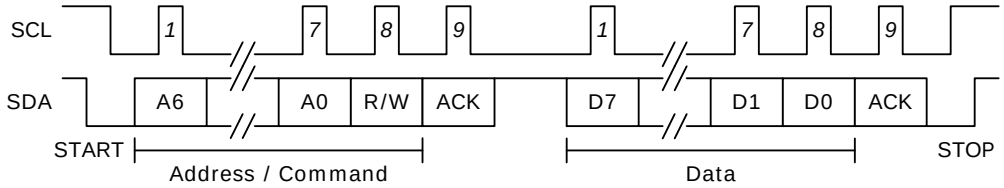
Setting CTRL.ENABLE enables the controller. The controller can be disabled by clearing the CTRL.ENABLE field, there is a chance that disabling is not allowed (at the time when it is attempted); the ENABLE_STATUS register shows if the controller was successful disabled.

Before enabling the controller, the user must decide on either standard or fast mode (CFG.SPEED) and configure clock dividers for generating the correct timing (SS_SCL_HCNT, SS_SCL_LCNT, FS_SCL_HCNT, FS_SCL_LCNT, and SDA_SETUP). The configuration of the divider registers depends on the VCore-III system clock frequency. The register descriptions explain how to calculate the required values.

Some two-wire serial devices requires a hold time on SDA after SCK when transmitting from the two-wire serial interface controller. The device supports a configurable hold delay through the TWI_CONFIG register.

The two-wire serial interface controller has an 8-byte combined receive and transmit FIFO.

*Figure 61 •* **Two-Wire Serial Interface Timing for 7-bit Address Access**




During normal operation of the two-wire serial interface controller, the STATUS register shows the activity and FIFO states.

### 5.8.3.1 Two-Wire Serial Interface Addressing

Use CFG.MASTER_10BITADDR and CFG.SLAVE_10BITADDR to configure either 7 or 10 bit addressing for master and slave modes respectively.

There are a number of reserved two-wire serial interface addresses. The two-wire serial interface controller does not restrict the use of these. However, if they are used out of context, there may be compatibility issues with other two-wire serial devices. The following table lists the two-wire serial interface reserved addresses.

***Table 134 •*** **Reserved Two-Wire Serial Interface Addresses**

| Register Address | Description |
|---|---|
| 0000 000 | General Call address/START Byte<br>If the slave is enabled the two-wire serial interface controller places the data in the receive buffer and issues a general call interrupt. The acknowledge response is configurable (through ACK_GEN_CALL). |
| 0000 001 | CBUS address. The two-wire serial interface controller ignores this address. |
| 0000 01X | Reserved, do not use. |
| 0000 1XX | Reserved, do not use. |
| 1111 1XX | Reserved, do not use. |
| 1111 0XX | 10-bit addressing indication, 7-bit address devices must not use this. |

The two-wire serial interface controller can general both General Call and START Byte. Initiate this through TAR.GC_OR_START_ENA or TAR.GC_OR_START. When operating as master, the target/slave address is configured using the TAR register.

### 5.8.3.2 Two-Wire Serial Interface Interrupt

The two-wire serial interface controller can generate a multitude of interrupts. All of these are described in the RAW_INTR_STAT register. The RAW_INTR_STAT register contains interrupt fields that are always set when their “trigger” conditions occur. The INTR_MASK register is used for masking interrupts and allowing interrupts to propagate to the INTR_STAT register. When set in the INTR_STAT register, the two-wire serial interface controller asserts interrupt toward the VCore-III interrupt controller.

The RAW_INTR_STAT register also specifies what is required to clear the specific interrupts. When the source of the interrupt is removed, reading the appropriate CLR_* register (for example, CLR_RX_OVER) clears the interrupt.

## 5.8.4 MII Management Controller

This section provides information about the MII Management controllers. The following table lists the registers associated with the MII Management controllers.

***Table 135 •*** **MIIM Registers**

| Register | Description |
|---|---|
| MII_STATUS | General configuration |
| MII_CMD | Target address |
| MII_DATA | Slave address |
| MII_CFG | Receive/transmit buffer and command |
| MII_SCAN_0 | Standard speed high time clock divider |
| MII_SCAN_1 | Standard speed low time clock divider |
| MII_SCAN_LAST_RSLTS | Fast speed high time clock divider |

*Table 135 •* **MIIM Registers**

| Register | Description |
|---|---|
| MII_SCAN_LAST_RSLTS_VLD | Fast speed low time clock divider |

The devices contain two MIIM controllers with equal functionality. Controller 0 is connected to the internal PHY, and controller 1 is used to manage external PHYs. Only the interface of controller 1 is available as pins on the device. Data is transferred on the MIIM interface using the Management Frame Format protocol specified in IEEE 802.3, Clause 22 or the MDIO Manageable Device protocol defined in IEEE 802.3, Clause 45. The clause 45 protocol differs from the clause 22 protocol by using indirect register accesses to increase the address range. The controller supports both Clause 22 and 45.

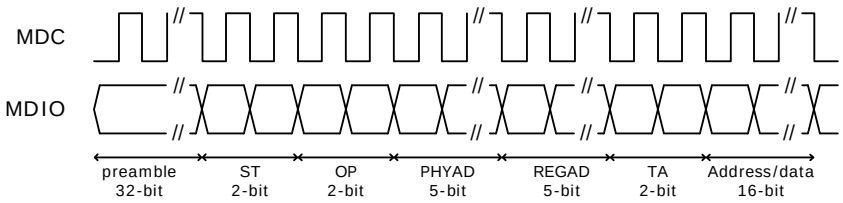
The following table lists the pins of the MIIM interface for controller 1.

*Table 136 •* **MIIM Management Controller Pins**

| Pin Name | I/O | Description |
|---|---|---|
| MDC | O | MIIM clock |
| MDIO | I/O | MIIM data input/output |

The MDIO signal is changed or sampled on the falling edge of the MDC clock by the controller. When the controller does not drive the MDIO pin it is tri-stated.

*Figure 62 •* **MII Management Timing**




### 5.8.4.1 Clock Configuration

The frequency of the management interface clock generated by the MIIM controller is derived from the VCore-III system frequency. The MIIM clock frequency is configurable and is selected with MII_CFG.MIIM_CFG_PRESCALE. The calculation of the resulting frequency is explained in the register description for MII_CFG.MIIM_CFG_PRESCALE. The maximum frequency of the MIIM clock is 25 MHz.

### 5.8.4.2 MII Management PHY Access

Reads and writes across the MII management interface are performed through the MII_CMD register. Details of the operation, such as the PHY address, the register address of the PHY to be accessed, the operation to perform on the register (for example, read or write), and write data (for write operations) are set in the MII_CMD register. When the appropriate fields of MII_CMD are set, the operation is initiated by writing 0x1 to MII_CMD.MIIM_CMD_VLD. The register is automatically cleared when the MIIM command is initiated. When initiating single MIIM commands, MII_CMD.MIIM_CMD_SCAN must be set to 0x0.

When an operation is initiated, the current status of the operation can be read in MII_STATUS. The fields MII_STATUS.MIIM_STAT_PENDING_RD and MII_STATUS.MIIM_STAT_PENDING_WR can be used to poll for completion of the operation. For a read operation, the read data is available in MII_DATA.MIIM_DATA_RDDATA after completion of the operation. The value of MII_DATA.MIIM_DATA_RDDATA is only valid if MII_DATA.MIIM_DATA_SUCCESS indicates no read errors.

The MIIM controller contains a small command FIFO. Additional MIIM commands can be queued as long as MII_STATUS.MIIM_STAT_OPR_PEND is cleared. Care must be taken with read operations, because multiple queued read operations will overwrite MII_DATA.MIIM_DATA_RDDATA.

**Note** A typical software implementation will never queue read operations, because the software needs read data before progressing the state of the software. In this case

MIIM_STATUS.MIIM_STAT_OPR_PEND is checked before issuing MIIM read or write commands, for read-operations MII_STATUS.MIIM_STAT_BUSY is checked before returning read result.

By default, the MIIM controller operates in clause 22 mode. To access clause 45 compatible PHYs, MII_CFG.MIIM_ST_CFG_FIELD and MII_CMD.MIIM_CMD_OPR_FIELD must be set according to clause 45 mode of operation.

#### 5.8.4.3 PHY Scanning

The MIIM controller can be configured to continuously read certain PHY registers and detect if the read value is different from an expected value. If a difference is detected, a special sticky bit register is set or a CPU interrupt is generated, or both. For example, the controller can be programmed to read the status registers of one or more PHYs and detect whether the Link Status changed since the sticky register was last read.

The reading of the PHYs is performed sequentially with the low and high PHY numbers specified in MII_SCAN_0 as range bounds. The accessed address within each of the PHYs is specified in MII_CMD.MIIM_CMD_REGAD. The scanning begins when a 0x1 is written to MII_CMD.MIIM_CMD_SCAN and a read operation is specified in MII_CMD.MIIM_CMD_OPR_FIELD. Setting MII_CMD.MIIM_CMD_SINGLE_SCAN stops the scanning after all PHYs have been scanned one time. The remaining fields of MII_CMD register is not used when scanning is enabled.

In MII_SCAN_1.MIIM_SCAN_EXPECT the expected value for the PHY register is set. The expected value is compared to the read value after applying the mask set in MII_SCAN_1.MIIM_SCAN_MASK. To “don’t care” a bit-position, write a 0 to the mask. If the expected value for a bit position differs from the read value during scanning, and the mask register has a 1 for the corresponding bit, a mismatch for the PHY is registered.

The scan results from the most recent scan can be read in MII_SCAN_LAST_RSLTS. The register contains one bit for each of the possible 32 PHYs. A mismatch during scanning is indicated by a 0. MII_SCAN_LAST_RSLTS_VLD will indicate for each PHY if the read operation performed during the scan was successful. The sticky-bit register MII_SCAN_RSLTS_STICKY has the mismatch bit set for all PHYs that had a mismatch during scanning since the last read of the sticky-bit register. When the register is read, its value is reset to all-ones (no mismatches).

#### 5.8.4.4 MII Management Interrupt

The MII management controllers can generate interrupts during PHY scanning. Each MII management controller has a separate interrupt signal to the interrupt controller. Interrupt is asserted when one or more PHYs have a mismatch during scan. The interrupt is cleared by reading the MII_SCAN_RSLTS_STICKY register, which resets all MII_SCAN_RSLTS_STICKY indications.

### 5.8.5 GPIO Controller

This section provides information about the use of GPIO pins.

The following table lists the registers associated with GPIO.

***Table 137 •*** **GPIO Registers**

| Register | Description |
|---|---|
| GPIO_OUT | Value to drive on GPIO outputs |
| GPIO_OUT_SET | Atomic set of bits in GPIO_OUT |
| GPIO_OUT_CLR | Atomic clear of bits in GPIO_OUT |
| GPIO_IN | Current value on the GPIO pins |
| GPIO_OE | Enable of GPIO output mode (drive GPIOs) |
| GPIO_ALT | Enable of overlaid GPIO functions |
| GPIO_INTR | Interrupt on changed GPIO value |
| GPIO_INTR_ENA | Enable interrupt on changed GPIO value |

*Table 137 •* **GPIO Registers *(continued)***

| Register | Description |
|---|---|
| GPIO_INTR_IDENT | Currently interrupting sources |

The GPIO pins are individually programmable. By default, GPIOs are inputs, however, they can be individually changed to outputs through GPIO_OE. For GPIOs that are in input mode, the value of the GPIO pin is reflected in the GPIO_IN register. GPIOs that are in output mode are driven to the value specified in GPIO_OUT.

In a system where multiple different CPU threads (or different CPUs) may work on the GPIOs at the same time, the GPIO_OUT_SET and GPIO_OUT_CLR registers provide a way for each thread to safely control the output value of GPIOs that are under their control, without having to implement locked regions and semaphores.

### 5.8.5.1 Overlaid Functions on the GPIOs

Most of the GPIO pins have overlaid (alternative) functions that can be enabled through the replicated GPIO_ALT register. For a particular GPIO *n*: Enable overlaid mode 1 by setting GPIO_ALT[0][n] and clearing GPIO_ALT[1][n]. Overlaid mode 2 is enabled by clearing GPIO_ALT[0][n] and setting GPIO_ALT[1][n]. For normal GPIO mode, clear both GPIO_ALT[0][n] and GPIO_ALT[1][n].

When the parallel interface is enabled (either master or slave mode), specific GPIO pins are overtaken and used for the parallel interface. This happens automatically when PI slave mode is enabled through the VCORE_CFG strapping pins or when the VCore-III CPU enables PI master mode through ICPU_CFG::GENERAL_CTRL.IF_MASTER_PI_ENA.

When the MIIM slave mode is enabled through the VCORE_CFG strapping pins, specific GPIO pins are overtaken and used for the MIIM slave interface. The PI master mode must not be enabled when MIIM slave mode is active.

*Table 138 •* **GPIO Mapping**

| GPIO Pin | Overlaid Function 1 | Overlaid Function 2 | MIIM Slave Interface |
|---|---|---|---|
| GPIO_0 | SIO_CLK | | |
| GPIO_1 | SIO_LD | | |
| GPIO_2 | SIO_DO | | |
| GPIO_3 | SIO_DI | | |
| GPIO_4 | TACHO | | |
| GPIO_5 | TWI_SCK | PHY0_LED1 | |
| GPIO_6 | TWI_SDA | PHY1_LED1 | |
| GPIO_7 | | PHY2_LED1 | |
| GPIO_8 | EXT_IRQ0 | PHY3_LED1 | |
| GPIO_9 | EXT_IRQ1 | PHY4_LED1 | |
| GPIO_10 | SFP14_SD | PHY5_LED1 | |
| GPIO_11 | SFP15_SD | PHY6_LED1 | |
| GPIO_12 | SFP17_SD | PHY7_LED1 | |
| GPIO_13 | SFP18_SD | PHY8_LED1 | |
| GPIO_14 | SI_nEn1 | PHY9_LED1 | SLV_ADDR |
| GPIO_15 | SI_nEn2 | PHY10_LED1 | SLV_MDC |
| GPIO_16 | SI_nEn3 | PHY11_LED1 | SLV_MDIO |

*Table 138 •* **GPIO Mapping *(continued)***

| GPIO Pin | Overlaid Function 1 | Overlaid Function 2 | MIIM Slave Interface |
|---|---|---|---|
| GPIO_17 | SFP10_SD | PHY0_LED0 | |
| GPIO_18 | SFP11_SD | PHY1_LED0 | |
| GPIO_19 | SFP12_SD | PHY2_LED0 | |
| GPIO_20 | SFP13_SD | PHY3_LED0 | |
| GPIO_21 | SFP16_SD | PHY4_LED0 | |
| GPIO_22 | SFP19_SD | PHY5_LED0 | |
| GPIO_23 | SFP24_SD | PHY6_LED0 | |
| GPIO_24 | SFP25_SD | PHY7_LED0 | |
| GPIO_25 | SFP20_SD | PHY8_LED0 | |
| GPIO_26 | SFP21_SD | PHY9_LED0 | |
| GPIO_27 | SFP22_SD | PHY10_LED0 | |
| GPIO_28 | SFP23_SD | PHY11_LED0 | |
| GPIO_29 | PWM | | |
| GPIO_30 | UART_TX | | |
| GPIO_31 | UART_RX | | |

For example, to enable the UART_RX and UART_TX overlaid functions, set bits 30 (enable UART_TX) and 31 (enable UART_RX) in the GPIO_ALT[0] register. The UART now has control of the GPIO pins.

### 5.8.5.2 GPIO Interrupt

The GPIO controller continually monitors all inputs and set bits in the GPIO_INTR register whenever a GPIO changes its input value. By enabling specific GPIO pins in the GPIO_INTR_ENA register, a change indication from GPIO_INTR is allowed to propagate (as GPIO interrupt) from the GPIO controller to the VCore-III Interrupt Controller.

The currently interrupting sources can be read from GPIO_INTR_IDENT, this register is the result of a binary AND between the GPIO_INTR and GPIO_INTR_ENA registers.

**Note** When the GPIO_INTR_IDENT register is different from zero, the GPIO controller is indicating an interrupt.

## 5.8.6 Serial GPIO Controller

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices feature a serial GPIO controller (SIO). By using a serial interface, the SIO controller significantly extends the number of available GPIOs with a minimum number of additional pins on the device. The main purpose of the SIO controller is to connect control signals from SFP modules; however, it can also act as an LED controller.

The SIO controller supports up to 128 serial GPIOs (SGPIOs) organized into 32 ports, with four SGPIOs per port. The following table lists the registers associated with the serial GPIO.

*Table 139 •* **SIO Registers**

| Register | Description | Replication |
|---|---|---|
| SIO_INPUT_DATA | Input data | SGPIOs per port (4) |
| SIO_INT_POL | Interrupt polarity | SGPIOs per port (4) |
| SIO_PORT_INT_ENA | Interrupt enable | None |
| SIO_PORT_CONFIG | Output port configuration | Per port (32) |

*Table 139 •* **SIO Registers *(continued)***

| Register | Description | Replication |
|---|---|---|
| SIO_PORT_ENABLE | Port enable | None |
| SIO_CONFIG | General configuration | None |
| SIO_CLOCK | Clock configuration | None |
| SIO_INT_REG | Interrupt register | SGPIOs per port (4) |

The following table lists the pins of the SIO controller. The pins of the SIO controller are overlaid on GPIOs. For more information about enabling the overlaid functionality of the GPIOs, see Overlaid Functions on the GPIOs, page 177.

*Table 140 •* **SIO Controller Pins**

| Pin Name | I/O | Description |
|---|---|---|
| SIO_CLK/GPIO_0 | O | SIO clock output, frequency is configurable using SIO_CLOCK.SIO_CLK_FREQ. |
| SIO_LD/GPIO_1 | O | SIO load data, polarity is configurable using SIO_CONFIG.SIO_LD_POLARITY. |
| SIO_DO/GPIO_2 | O | SIO data output. |
| SIO_DI/GPIO_3 | I | SIO data input. |

The SIO controller works by shifting SGPIO values out on SIO_DO though a chain of shift registers on the PCB. After shifting a configurable number of SGPIO bits, the SIO controller asserts SIO_LD, which causes the shift registers to apply the values of the shifted bits to outputs. The SIO controller is also capable of reading inputs, at the same time as shifting out SGPIO values on SIO_DO, it also samples the SIO_DI input. The values sampled on SIO_DI are made available to software.

If the SIO controller is only used for outputs, the use of the load signal is optional. If the load signal is omitted, simpler shift registers (without load) can be used, however, the outputs of these registers will toggle during shifting.

When driving LED outputs, it is acceptable that the outputs will toggle when SGPIO values are updated (shifted through the chain). When the shift frequency is fast, the human eye is not able to see the shifting though the LEDs.

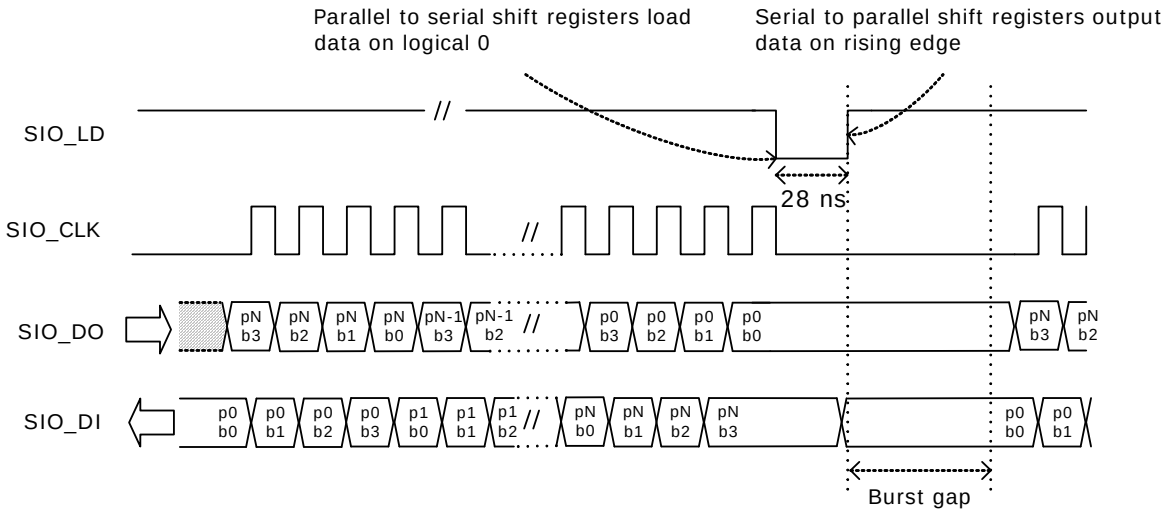
The number of shift registers in the chain is configurable. The SIO controller allows enabling of individual ports through SIO_PORT_ENABLE; only enabled ports are shifted out on SI_DO. Ports that are not enabled are skipped during shifting of GPIO values.

**Note** SIO_PORT_ENABLE allows skipping of ports in the SGPIO output stream that are not in use. The number of GPIOs per (enabled) port is configurable as well, through SIO_CONFIG.SIO_PORT_WIDTH this can be set to 1,2,3, or 4 bits. The number of bits per port is common for all enabled ports, so the number of shift registers on the PCB must be equal to the number of enabled ports times the number of SGPIOs per port.

Enabling of ports and configuration of SGPIOs per port applies to both output mode and input mode. Unlike a regular GPIO port, a single SGPIO position can be used both as output and input. That is, software can control the output of the shift register AND read the input value at the same time. Using SGPIOs as inputs requires load-capable shift registers.

Regular shift registers and load-capable shift-registers can be mixed, which is useful when driving LED indications for integrated PHYs at the same time as supporting reading of link status from SFP modules, for example.

***Figure 63 •*** **SIO Timing**



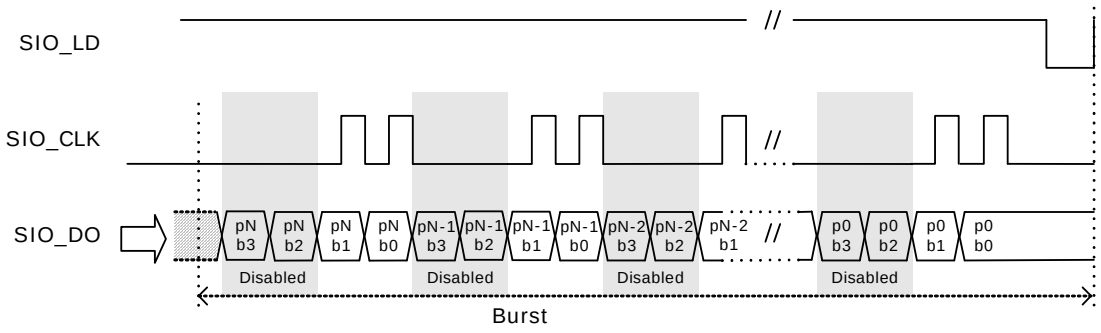

The SGPIO values are output in bursts followed by assertion of the SIO_LD signal. Values can be output as a single burst, or as continuous bursts separated by a configurable burst gap. The maximum length of a burst is 32 × 4 data cycles. The burst gap is configurable in steps of approximately 1 ms between 0 ms and 33 ms through SIO_CONFIG.SIO_BURST_GAP_DIS and SIO_CONFIG.SIO_BURST_GAP.

A single burst is issued by setting SIO_CONFIG.SIO_SINGLE_SHOT. The field is automatically cleared by hardware when the burst is finished. To issue continuous bursts, set SIO_CONFIG.SIO_AUTO_REPEAT. The SIO controller continues to issue bursts until SIO_CONFIG.SIO_AUTO_REPEAT is cleared.

SGPIO output values are configured in SIO_PORT_CONFIG.BIT_SOURCE. The input value is available in SIO_INPUT_DATA.S_IN.

The following illustration shows what happens when the number of SGPIOs per port is configured to 2 (through SIO_CONFIG.SIO_PORT_WIDTH). Disabling of ports (through SIO_PORT_ENABLE) is handled in the same way as disabling the SGPIO ports.

***Figure 64 •*** **SIO Timing with SGPIOs Disabled**




The frequency of the SIO_CLK clock output is configured through SIO_CLOCK.SIO_CLK_FREQ. The SIO_LD output is asserted after each burst, this output is asserted for 28 ns. The polarity of SIO_LD is configurable through SIO_CONFIG.SIO_LD_POLARITY.

The SIO_LD output can be used to ensure that outputs are stable when serial data is being shifted through the registers. This can be done by using the SIO_LD output to shift the output values into serial-to-parallel registers after the burst is completed. If serial-to-parallel registers are not used, the outputs will toggle while the burst is being shifted through the chain of shift registers. A universal serial-to-parallel shift register outputs the data on a positive-edge load signal, and a universal parallel-to-serial shift

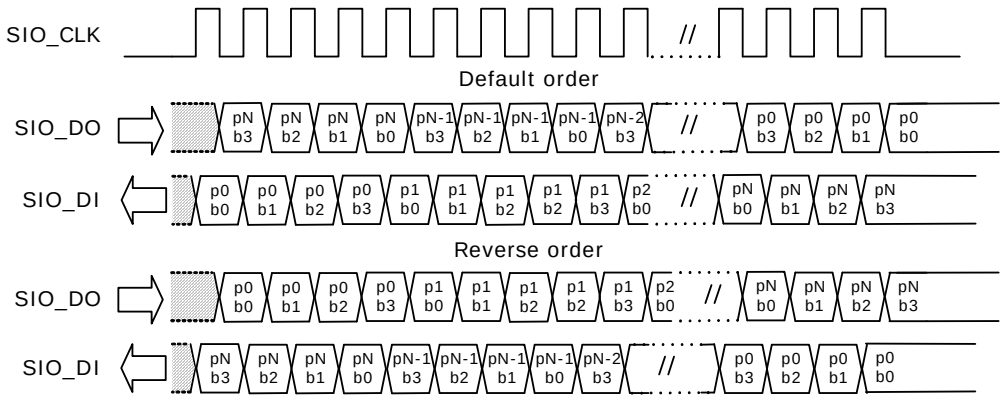
register shifts data when the load pin is high, so one common load signal can be used for both input and output serial <-> parallel conversion.

The assertion of SIO_LD happens after the burst to ensure that after power up, the single burst will result in well-defined output registers. Consequently, to sample input values one time, two consecutive bursts must be issued. The first burst results in the input values being sampled by the serial-to-parallel registers, and the second burst shifts the input values into the SIO controller.

The required port order in the serial bitstream depends on the physical layout of the shift register chain. Often the input and output port orders must be opposite in the serial streams. The port order of the input and output bitstream is independently configurable in SIO_CONFIG.SIO_REVERSE_INPUT and SIO_CONFIG.SIO_REVERSE_OUTPUT.

The following illustration shows the port order.

***Figure 65 •*** **SIO Output Order**




### 5.8.6.1 Output Modes

The output mode of each SGPIO can be individually configured in SIO_PORT_CONFIG.BIT_SOURCE. The SIO controller features three output modes:

- Static
- Blink
- Link activity

**Static Mode** The static mode is used to assign a fixed value to the SGPIO, for example, fixed 0 or fixed 1.

**Blink Mode** The blink mode makes the SGPIO blink at a fixed rate. The SIO controller features two blink modes that can be set independently. A SGPIO can then be configured to use either blink mode 0 or blink mode 1. The blink outputs are configured in SIO_CONFIG.SIO_BMODE_0 and SIO_CONFIG.SIO_BMODE_1. To synchronize the blink modes between different devices, reset the blink counter using SIO_CONFIG.SIO_BLINK_RESET. The “burst toggle” mode of blink mode 1 toggles the output with every burst.

***Table 141 •*** **Blink Modes**

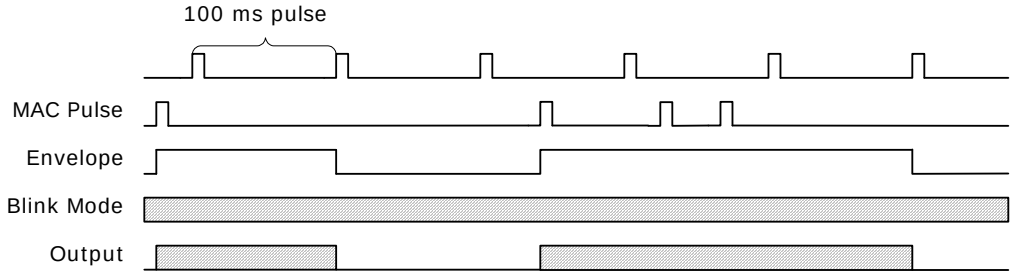
| Mode | Description |
|---|---|
| Blink mode 0 | 0: 20 Hz blink frequency<br>1: 10 Hz blink frequency<br>2: 5 Hz blink frequency<br>3: 2.5 Hz blink frequency |
| Blink mode 1 | 0: 20 Hz blink frequency<br>1: 10 Hz blink frequency<br>2: 5 Hz blink frequency<br>3: Burst toggle |

**Link Activity Mode** The link activity mode makes the output blink when there is activity on the port module (Rx or Tx). The mapping between SIO port number port module number is 1:1, For example, port 0 is connected to port module 0, port 1 is connected to port module 1, and so on.

The link activity mode uses an envelope signal to gate the selected blinking pattern (blink mode 0 or blink mode 1). When the envelope signal is asserted, the output blinks, and when the envelope pattern is de-asserted, the output is turned off. To ensure that even a single packet makes a visual blink, an activity pulse from the port module is extended to minimum 100 ms. If another packet is sent while the envelope signal is asserted, the activity pulse is extended by another 100 ms. The polarity of the link activity modes can be set in SIO_PORT_CONFIG.BIT_SOURCE.

The following illustration shows the link activity timing.

***Figure 66 •*** **Link Activity Timing**




### 5.8.6.2 SIO Interrupt

The SIO controller can generate interrupts based on the value of the input value of the SGPIOs. All interrupts are level sensitive.

Interrupts are enabled using the two registers. Interrupts can be individually enabled for each port in SIO_PORT_INT_ENA.INT_ENA (32 bits) and in SIO_CONFIG.SIO_INT_ENA (4 bits) interrupts are enabled for the four inputs per port. In other words, SIO_CONFIG.SIO_INT_ENA is common for all 32 ports. The polarity of interrupts is configured for each SGPIO in SIO_INT_POL.

The SIO controller has one interrupt output connected to the main interrupt controller, which is asserted when one or more interrupts are active. To determine which SGPIO is causing the interrupt, the CPU must read the sticky bit interrupt register SIO_INT_REG. The register has one bit per SGPIO and can only be cleared by software. A bit is cleared by writing a 1 to the bit position. The interrupt output remains high until all interrupts in SIO_INT_REG are cleared.

### 5.8.6.3 Loss of Signal Detection

The SIO controller can propagate loss of signal detection inputs directly to the signal detection input of the port modules. This is useful when, for example, SFP modules are connected to the device. The mapping between SIO ports and port modules is the same as for the link activity inputs; port 0 is connected to port module 0, port1 is connected to port module 1, and so on.

The value of SGPIO bit 0 of each SIO port is forwarded directly to the loss of signal input on the corresponding device. The device must enable the loss of signal input locally in the device.

Loss of signal can also be taken directly from overlaid functions on the regular GPIOs. When that is the case the input from the SIO controller is ignored. For more information, see Overlaid Functions on the GPIOs, page 177.

The polarity of the loss of signal input is configured using SIO_INT_POL, meaning the same polarity must be used for loss of signal detect and interrupt.

## 5.8.7 FAN Controller

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices include a fan controller that can be used to control and monitor a system fan. The fan speed is regulated using a pulse-width-modulation (PWM) output. The fan speed is monitored using a TACHO input. This is especially powerful when combined with the internal temperature sensor (in the PHY).

The following table lists the registers associated with the fan controller.

***Table 142 •*** **Fan Controller Registers**

| Register | Description |
|---|---|
| FAN_CFG | General configuration |
| FAN_CNT | Fan revolutions counter |

The following table lists the pins of the fan controller. The pins of the fan controller are overlaid on GPIOs. For more information about enabling the overlaid functionality of GPIOs, see Overlaid Functions on the GPIOs, page 177.

***Table 143 •*** **Fan Controller Pins**

| Pin Name | I/O | Description |
|---|---|---|
| TACHO/GPIO_4 | I | TACHO input for counting revolutions. |
| PWM/GPIO_29 | O | PWM fan output. |

The PWM output can be configured to any of the following frequencies in FAN_CFG.PWM_FREQ:

- 10 Hz
- 20 Hz
- 40 Hz
- 60 Hz
- 80 Hz
- 100 Hz
- 120 Hz
- 25 kHz

The low frequencies can be used for driving three-wire fans using a FET/transistor. The 25 kHz frequency can be used for four-wire fans that use the PWM input internally to control the fan. The duty cycle of the PWM output is programmable from 0% to 100%, with 8-bit accuracy. The polarity of the output can be controlled by FAN_CFG.INV_POL, so a duty-cycle of 100%, for example, can be either always low or always high.

The PWM output pin can be configured to act as a normal output or as an open-collector output, where the output value of the pin is kept low, but the output enable is toggled. The open-collector output mode is enabled by setting FAN_CFG.PWM_OPEN_COL_ENA.

**Note** By using open-collector mode, it is possible to do external pull-up to higher voltage than the maximum GPIO I/O supply. The GPIOs are 5V-tolerable.

The speed of the fan can be measured using a 16-bit wrapping counter that counts the rising edges on the TACHO-input. A fan usually gives 1-4 pulses per revolution depending on the fan type. Optionally, the TACHO-input can be gated by the polarity-corrected PWM output by setting FAN_CFG.GATE_ENA, so that only TACHO pulses received while the polarity corrected PWM output is high are counted. Glitches on the TACHO-input can occur right after the PWM output goes high, therefore the gate signal is delayed by 10 µs when PWM goes high. There is no delay when PWM goes low, and the length of the delay is not configurable. Software reads the counter value in FAN_CNT and calculates the RPM of the fan.

The following is an example of how to calculate the RPM of the fan: If the fan controller is configured to 100 Hz and a 20% duty cycle, each PWM pulse is high in 2 ms and low in 8 ms. If gating is enabled the gating of the TACHO-input is “open” in 1.99 ms and “closed” in 8.01 ms. If the fan is turning with 100 RPM and gives two TACHO pulses per revolution, it will ideally give 200 pulses per minute. TACHO pulses are only counted in 19.99% of the time, so it will give 200× 0.1999 = 39.98 pulses per minute. If the additional 10 µs gating time is ignored, the counter value is multiplied by 5/2 to get the RPM value, because there is a 20% duty cycle with two TACHO pulses per revolution. By multiplying with 5/2, the RPM value is calculated to 99.95, which is 0.05% off the correct value (due to the 10 µs gating time).

## 5.8.8 Interrupt Controller

This section provides information about the VCore-III interrupt controller.

The following table lists the registers associated with the interrupt controller.

***Table 144 •*** **Interrupt Controller Registers**

| Register | Description |
|---|---|
| **Configuration and status for interrupts** | |
| ICPU_IRQ0_ENA | Global enable of ICPU_IRQ0 interrupt |
| ICPU_IRQ0_IDENT | Currently interrupting ICPU_IRQ0 sources |
| ICPU_IRQ1_ENA | Global enable of ICPU_IRQ1 interrupt |
| ICPU_IRQ1_IDENT | Currently interrupting ICPU_IRQ1 sources |
| EXT_IRQ0_ENA | Global enable of EXT_IRQ0 interrupt |
| EXT_IRQ0_IDENT | Currently interrupting EXT_IRQ0 sources |
| EXT_IRQ1_ENA | Global enable of EXT_IRQ1 interrupt |
| EXT_IRQ1_IDENT | Currently interrupting EXT_IRQ1 sources |
| **Configuration of individual interrupt sources** | |
| EXT_IRQ0_INTR_CFG | EXT_IRQ0 source configuration |
| EXT_IRQ1_INTR_CFG | EXT_IRQ1 source configuration |
| SW0_INTR_CFG | SW0 source configuration |
| SW1_INTR_CFG | SW1 source configuration |
| PI_SD0_INTR_CFG | PI_SD0 source configuration |
| PI_SD1_INTR_CFG | PI_SD1 source configuration |
| UART_INTR_CFG | UART source configuration |
| TIMER0_INTR_CFG | TIMER0 source configuration |
| TIMER1_INTR_CFG | TIMER1 source configuration |
| TIMER2_INTR_CFG | TIMER2 source configuration |
| FDMA_INTR_CFG | FDMA source configuration |
| TWI_INTR_CFG | TWI source configuration |
| GPIO_INTR_CFG | GPIO source configuration |
| SGPIO_INTR_CFG | SGPIO source configuration |
| DEV_ALL_INTR_CFG | DEV_ALL source configuration |
| XTR_RDY0_INTR_CFG | XTR_RDY0 source configuration |
| XTR_RDY1_INTR_CFG | XTR_RDY1 source configuration |
| INJ_RDY0_INTR_CFG | INJ_RDY0 source configuration |
| INJ_RDY1_INTR_CFG | INJ_RDY1 source configuration |
| PTP_SYNC_INTR_CFG | PTP_SYNC source configuration |
| MIIM0_INTR_CFG | MIIM0 source configuration |
| MIIM1_INTR_CFG | MIIM1 source configuration |
| **General enable/disable and status for all interrupt sources** | |
| INTR | Interrupt sticky bits |

*Table 144 •* **Interrupt Controller Registers *(continued)***

| Register | Description |
|---|---|
| INTR_ENA | Interrupt enable |
| INTR_ENA_SET | Atomic set of bits in INTR_ENA |
| INTR_ENA_CLR | Atomic clear of bits in INTR_ENA |
| INTR_RAW | Raw value of interrupt from sources |
| DEV_IDENT | Currently interrupting DEV_ALL sources |

Possible sources of the DEV_ALL interrupt are:

- Fast link status from the PHYs for port 0 through 11 (DEV_IDENT[11:0])
- PCS link status from the PCS for port 12 through 25 (DEV_IDENT[25:12])
- PCS link status from the PCS for port 10 (DEV_IDENT[26])
- PCS link status from the PCS for port 11 (DEV_IDENT[27])
- Global PHY interrupt (DEV_IDENT[28])

Each of the interrupt sources in the VCore-III system can be individually assigned to one of four possible interrupt outputs: Two ICPU_IRQ interrupt outputs go directly to the VCore-III CPU, and two EXT_IRQ interrupt allow interrupting external devices.

Each interrupt output has a global enable register, ICPU_IRQ0_ENA, ICPU_IRQ1_ENA, EXT_IRQ0_ENA, and EXT_IRQ1_ENA. This register must be set in order for the interrupt outputs to propagate interrupts. When there is an active interrupt on any interrupt output, the ICPU_IRQ0_IDENT, ICPU_IRQ1_IDENT, EXT_IRQ0_IDENT, and EXT_IRQ1_IDENT registers show the active interrupt sources for each individual interrupt.

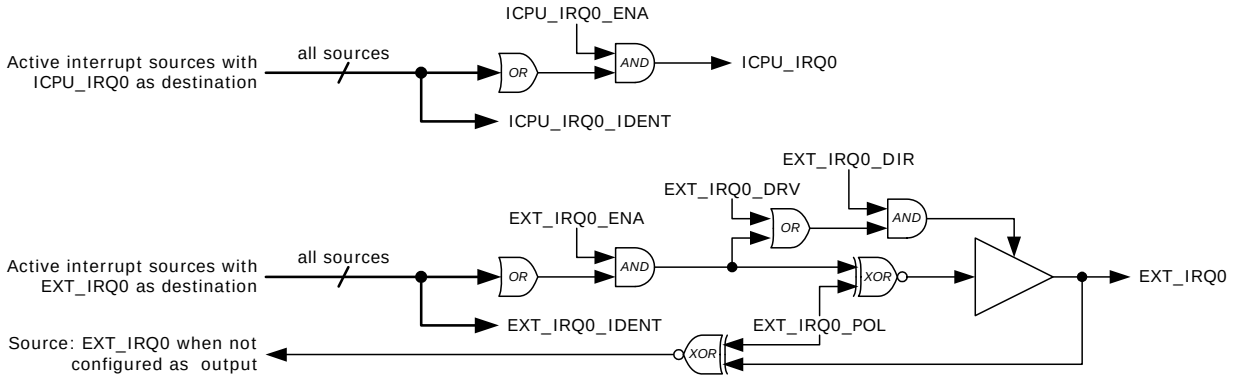
The two EXT_IRQ0 pins are special, because they are overlaid functions on the GPIO interface. The active level of the EXT_IRQ pins is configured individually through the INTR_POL field of EXT_IRQ0_INTR_CFG and EXT_IRQ1_INTR_CFG. Additionally, the EXT_IRQ pins operate as either interrupt outputs or as interrupt sources. This is individually configured through the INTR_DIR field of EXT_IRQ0_INTR_CFG and EXT_IRQ1_INTR_CFG. When operating as outputs, the EXT_IRQ pins can be tri-stated when there is no interrupt. This is configured through the field INTR_DRV in EXT_IRQ0_INTR_CFG and EXT_IRQ1_INTR_CFG.

For more information about the location on the GPIOs and how to enable the overlaid function, see GPIO Controller, page 176.

When an interrupt output is configured to drive only during interrupt, interrupt outputs from multiple devices can be connected in parallel with a pull-resistor to make wired-or/and interrupts. EXT_IRQ0_INTR_CFG and EXT_IRQ1_INTR_CFG (or both) must be configured before enabling the overlaid GPIO functions.

The following illustration depicts only ICPU_IRQ0 and EXT_IRQ0. ICPU_IRQ1 and EXT_IRQ1 is similar, except zeros replace the ones.

***Figure 67 •*** **Logical Equivalent for Interrupt Outputs**




*Note Internally in the device, all interrupt sources are active high.*

Each interrupt source has its own configuration register (*_INTR_CFG). The sticky functionality can be bypassed by means of the INTR_BYPASS field. For software development, an interrupt event can be emulated by setting the one-shot INTR_FORCE field. The destination interrupt output is configured through the INTR_SEL field. Interrupt outputs can have many sources, but each source can only have one destination.

The bypass feature can be useful when only a single, or just a few, interrupt source is enabled for a specific interrupt output. When stickiness in the interrupt controller is bypassed, clearing the interrupt indication at its source also clears the associated interrupt.

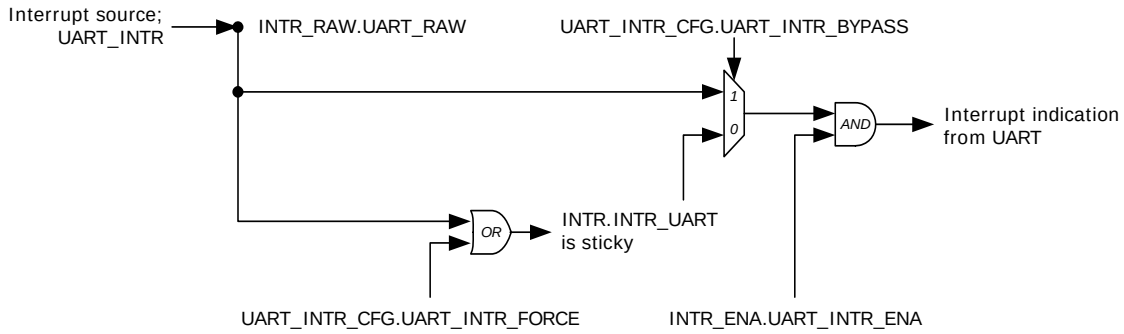
If an interrupt source indicates an interrupt, the associated field in the INTR register is set, this is a sticky indication. The current interrupt inputs from the sources are available through INTR_RAW.

For an interrupt to propagate to its destination, it must be enabled by setting the associated INTR_ENA field. In a system where multiple different CPU threads (or different CPUs) may work on the interrupts at the same time, the INTR_ENA_SET and INTR_ENA_CLR registers provide a method for each thread to safely control enabling and disabling of the interrupts that are under their control, without having to implement locked regions and semaphores.

The following illustration shows an example of the UART interrupt; however, it is representative to any other interrupt by substituting UART for the interrupt name.

The timer interrupt sources are only asserted for a single clock cycle (when the timer wraps). As a result, the trigger and bypass functions (as depicted) are not needed (nor implemented) for the timer interrupt sources.

***Figure 68 •*** **Logical Equivalent for Interrupt Sources**



# 6 Features

This section provides information about specific features supported by individual blocks in the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices and describes how these features are administrated by configurations across the entire device. Examples of various standard features are described such as the support for different spanning tree versions and VLAN operations, and more advanced features, such as QoS and VCAP.

## 6.1 Port Mapping

This section provides information about the mapping from switch core port modules to SerDes type to physical interface pins on the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

When accessing port module registers (PORT::), port masks in the analyzer, or in general, whenever a switch core register refers to a port, the internal switch port module number must be used.

### 6.1.1 VSC7424-02 Port Mapping

The internal port modules in the switch core maps to external pins on the VSC7424-02 device for the two switch modes according to the following table.

***Table 145 •*** **VSC7424-02: Mapping from Port Modules to Physical Interface Pins**

| Port Number | Switch Port Module | Interface Type | SerDes Type | Interface Pins |
|---|---|---|---|---|
| 0 – 7 | 0 – 7 | Internal PHY | | P*x*_D[3:0]N, P*x*_D[3:0]P, where *x* is 0 through 7 |
| 8 | 24 | 1G SGMII | SERDES1G | SerDes0_TxP, SerDes0_TxN, SerDes0_RxP, SerDes0_RxN |
| 9 | 25 | 1G SGMII | SERDES1G | SerDes1_TxP, SerDes1_TxN, SerDes1_RxP, SerDes1_RxN |
| | 26 | CPU port | | |

### 6.1.2 VSC7425-02 Port Mapping

VSC7425-02 has the option to run in one of two switch modes controlling the type and number of external Ethernet interfaces:

- Switch mode 0 enables 12× CuPHY + 1× QSGMII + 2× 1G SGMII
- Switch mode 2 enables 12× CuPHY + 6× 1G SGMII

The switch mode is controlled through DEVCPU_GCB::MISC_CFG.SW_MODE.

The internal port modules in the switch core maps to external pins on the VSC7424 device for the two switch modes according to the following tables.

***Table 146 •*** **VSC7425-02 Switch Mode 0: Mapping from Port Modules to Physical Interface Pins**

| Port Number | Switch Port Module | Interface Type | SerDes Type | Interface Pins |
|---|---|---|---|---|
| 0 – 11 | 0 – 11 | Internal PHY | | P*x*_D[3:0]N, P*x*_D[3:0]P, where *x* is 0 through 11 |
| 12 – 15 | 20 – 23 | QSGMII | SERDES6G | SerDes_E1_TxP, SerDes_E1_TxN, SerDes_E1_RxP, SerDes_E1_RxN |
| 16 | 24 | 1G SGMII | SERDES1G | SerDes0_TxP, SerDes0_TxN, SerDes0_RxP, SerDes0_RxN |

*Table 146* • **VSC7425-02 Switch Mode 0: Mapping from Port Modules to Physical Interface Pins**

| Port Number | Switch Port Module | Interface Type | SerDes Type | Interface Pins |
|---|---|---|---|---|
| 17 | 25 | 1G SGMII | SERDES6G | SerDes_E0_TxP, SerDes_E0_TxN, SerDes_E0_RxP, SerDes_E0_RxN |

*Table 147* • **VSC7425-02 Switch Mode 2: Mapping from Port Modules to Physical Interface Pins**

| Port Number | Switch Port Module | Interface Type | SerDes Type | Interface Pins |
|---|---|---|---|---|
| 0 – 11 | 0 – 11 | Internal PHY | | P*x*_D[3:0]N, P*x*_D[3:0]P, where *x* is 0 through 11 |
| 12 | 20 | 1G SGMII | SERDES1G | SerDes5_TxP, SerDes5_TxN, SerDes5_RxP, SerDes5_RxN |
| 13 | 21 | 1G SGMII | SERDES1G | SerDes4_TxP, SerDes4_TxN, SerDes4_RxP, SerDes4_RxN |
| 14 | 22 | 1G SGMII | SERDES1G | SerDes3_TxP, SerDes3_TxN, SerDes3_RxP, SerDes3_RxN |
| 15 | 23 | 1G SGMII | SERDES1G | SerDes2_TxP, SerDes2_TxN, SerDes2_RxP, SerDes2_RxN |
| 16 | 24 | 1G SGMII | SERDES1G | SerDes0_TxP, SerDes0_TxN, SerDes0_RxP, SerDes0_RxN |
| 17 | 25 | 1G SGMII | SERDES1G | SerDes1_TxP, SerDes1_TxN, SerDes1_RxP, SerDes1_RxN |
| | 26 | CPU port | | |

## 6.1.3 VSC7426-02 Port Mapping

The internal port modules in the switch core maps to external pins on the VSC7426 device as shown in the following table.

*Table 148* • **VSC7426-02: Mapping from Port Modules to Physical Interface Pins**

| Port Number | Switch Port Module | Interface Type | SerDes Type | Interface Pins |
|---|---|---|---|---|
| 0 – 11 | 0 – 11 | Internal PHY | | P*x*_D[3:0]N, P*x*_D[3:0]P, where *x* is 0 through 11 |
| 12 – 15 | 12 – 15 | QSGMII | SERDES6G | SerDes_E3_TxP, SerDes_E3_TxN, SerDes_E3_RxP, SerDes_E3_RxN |
| 16 – 19 | 16 – 19 | QSGMII | SERDES6G | SerDes_E2_TxP, SerDes_E2_TxN, SerDes_E2_RxP, SerDes_E2_RxN |
| 20 – 23 | 20 – 23 | QSGMII | SERDES6G | SerDes_E1_TxP, SerDes_E1_TxN, SerDes_E1_RxP, SerDes_E1_RxN |
| | 26 | CPU port | | |

### 6.1.4 VSC7427-02 Port Mapping

The internal port modules in the switch core maps to external pins on the VSC7427 device as shown in the following table.

***Table 149 •*** **VSC7427-02: Mapping from Port Modules to Physical Interface Pins**

| Port Number | Switch Port Module | Interface Type | SerDes Type | Interface Pins |
|---|---|---|---|---|
| 0 – 11 | 0 – 11 | Internal PHY | | P*x*_D[3:0]N, P*x*_D[3:0]P, where *x* is 0 through 11 |
| 12 – 15 | 12 – 15 | QSGMII | SERDES6G | SerDes_E3_TxP, SerDes_E3_TxN, SerDes_E3_RxP, SerDes_E3_RxN |
| 16 – 19 | 16 – 19 | QSGMII | SERDES6G | SerDes_E2_TxP, SerDes_E2_TxN, SerDes_E2_RxP, SerDes_E2_RxN |
| 20 – 23 | 20 – 23 | QSGMII | SERDES6G | SerDes_E1_TxP, SerDes_E1_TxN, SerDes_E1_RxP, SerDes_E1_RxN |
| 24 | 24 | 1G SGMII | SERDES1G | SerDes0_TxP, SerDes0_TxN, SerDes0_RxP, SerDes0_RxN |
| 25 | 25 | 1G SGMII | SERDES6G | SerDes_E0_TxP, SerDes_E0_TxN, SerDes_E0_RxP, SerDes_E0_RxN |
| | 26 | CPU port | | |

## 6.2 Switch Control

This section provides information about the minimum requirements for switch operation.

### 6.2.1 Switch Initialization

The following initialization sequence is required to ensure proper operation of the switch:

1. Configure the desired switch mode in DEVCPU_GCB::MISC_CFG.SW_MODE.
2. Initialize memories:
   SYS.RESET_CFG.MEM_ENA = 1.
   SYS.RESET_CFG.MEM_INIT = 1.
3. Wait 100 µs for memories to initialize (SYS.RESET_CFG.MEM_INIT cleared).
4. Enable the switch core:
   SYS.RESET_CFG.CORE_ENA = 1.
5. Release reset of the internal PHYs:
   DEVCPU_GCB.SOFT_CHIP_RST.SOFT_PHY_RST = 0.
6. Enable each port module through SYS.PORT.SWITCH_PORT_MODE.PORT_ENA = 1.

## 6.3 Port Module Control

This section provides information about the features and configurations for port control, port reset procedures, and port counters.

### 6.3.1 MAC Configuration Port Mode Control

All port modules can be configured independently to the speed and duplex modes listed in the following tables.

***Table 150 •*** **MAC Configuration of Port Modes for Ports with Internal PHYs**

| Configuration | 10 Mbps, Half Duplex | 10 Mbps, Full Duplex | 100 Mbps, Half Duplex | 100 Mbps, Full Duplex | 1 Gbps, Full Duplex |
|---|---|---|---|---|---|
| PORT::CLOCK_CFG.LINK_SPEED | | | | | |

*Table 150 •* **MAC Configuration of Port Modes for Ports with Internal PHYs *(continued)***

| Configuration | 10 Mbps, Half Duplex | 10 Mbps, Full Duplex | 100 Mbps, Half Duplex | 100 Mbps, Full Duplex | 1 Gbps, Full Duplex |
|---|---|---|---|---|---|
| PORT::MAC_MODE_CFG.FDX_ENA | 0 | 1 | 0 | 1 | 1 |
| PORT::MAC_MODE_CFG.GIGA_MODE_ENA | 0 | 0 | 0 | 0 | 1 |
| SYS:PORT:FRONT_PORT_MODE.HDX_MODE | 1 | 0 | 1 | 0 | 0 |
| PORT::MAC_IFG_CFG.TX_IFG | 17 | 17 | 17 | 17 | 5 |
| PORT::MAC_IFG_CFG.RX_IFG1 | 11 | | 11 | | |
| PORT::MAC_IFG_CFG.RX_IFG2 | 9 | | 9 | | |
| PORT::MAC_HDX_CFG.LATE_COL_POS | 64 | | 64 | | |
| SYS:PORT:FRONT_PORT_MODE.HDX_MODE | 1 | 0 | 1 | 0 | 0 |

*Table 151 •* **MAC Configuration of Port Modes for Ports with SerDes**

| Configuration | 10 Mbps, Half Duplex | 10 Mbps, Full Duplex | 100 Mbps, Half Duplex | 100 Mbps, Full Duplex | 1 Gbps, Full Duplex |
|---|---|---|---|---|---|
| PORT::CLOCK_CFG.LINK_SPEED | 3 | 3 | 2 | 2 | 1 |
| PORT::MAC_MODE_CFG.FDX_ENA | 0 | 1 | 0 | 1 | 1 |
| PORT::MAC_MODE_CFG.GIGA_MODE_ENA | 0 | 0 | 0 | 0 | 1 |
| SYS:FRONT_PORT_MODE.HDX_MODE | 1 | 0 | 1 | 0 | 0 |
| PORT::MAC_IFG_CFG.TX_IFG | 15 | 15 | 15 | 15 | 5 |
| PORT::MAC_IFG_CFG.RX_IFG1 | 11 | | 7 | | |
| PORT::MAC_IFG_CFG.RX_IFG2 | 9 | | 9 | | |
| PORT::MAC_HDX_CFG.LATE_COL_POS | 67 | | 67 | | |
| SYS:FRONT_PORT_MODE.HDX_MODE | 1 | 0 | 1 | 0 | 0 |

## 6.3.2 SerDes Configuration Port Mode Control

Each SerDes port can connect to one of two types of SerDes macros. Ports connecting to SERDES6G must be configured according to the following table.

*Table 152 •* **SERDES6G Configuration**

| Configuration | SGMII Mode | QSGMII Mode |
|---|---|---|
| hsio::serdes6g_pll_cfg.pll_rot_frq | 0 | 0 |
| hsio::serdes6g_pll_cfg.pll_rot_dir | 1 | 0 |
| hsio::serdes6g_pll_cfg.pll_ena_rot | 0 | 0 |
| hsio::serdes6g_common_cfg.ena_lane | 1 | 1 |
| hsio::serdes6g_common_cfg.if_mode | 1 | 3 |
| hsio::serdes6g_common_cfg.qrate | 1 | 0 |
| hsio::serdes6g_common_cfg.hrate | 0 | 0 |
| hsio::serdes6g_ib_cfg1.ib_reserved | 1 | 1 |

Ports connecting to a SERDES1G must be configured according to the following table.

***Table 153 •*** **SERDES1G Configuration**

| Configuration | SGMII mode |
|---|---|
| hsio::serdes1g_common_cfg.ena_lane | 1 |

## 6.3.3 Port Reset Procedure

When changing a switch port's mode of operation or restarting a switch port, the following port reset procedure must be followed:

1. Disable the MAC frame reception in the switch port:
   PORT::MAC_ENA_CFG.RX_ENA = 0.
2. Disable traffic being sent to or from the switch port:
   SYS:PORT:SWITCH_PORT_MODE_ENA = 0
   SYS:PORT:FRONT_PORT_MODE_HDX_MODE = 0.
3. Disable shaping to speed up flushing of frames
   SYS:SCH_SHAPING_CTRL.PORT_SHAPING_ENA = 0,
   SYS:SCH_SHAPING_CTRL.PRIO_SHAPING_ENA = 0.
4. Flush the queues associated with the port:
   REW:PORT:PORT_CFG.FLUSH_ENA = 1.
5. Wait at least the time it takes to receive a frame of maximum length on the port Worst-case delays for 10 kilobyte jumbo frames are:
   8 ms on a 10M port
   800 µs on a 100M port
   80 µs on a 1G port.
6. Reset the switch port by setting the following reset bits in CLOCK_CFG:
   PORT::CLOCK_CFG.MAC_TX_RST = 1,
   PORT::CLOCK_CFG.MAC_RX_RST = 1,
   PORT::CLOCK_CFG.PORT_RST = 1,
   PORT::CLOCK_CFG.PHY_RST = 1 (if port is connected to an internal PHY).
7. Wait until flushing is complete:
   SYS:PORT:SW_STATUS.EQ_AVAIL must return 0.
8. Clear flushing again:
   REW:PORT:PORT_CFG.FLUSH_ENA = 0.
9. Re-enable traffic being sent to or from the switch port:
   SYS:PORT:SWITCH_PORT_MODE.PORT_ENA = 1.
10. Set up the switch port to the new mode of operation. Keep the reset bits in CLOCK_CFG set. For more information about port mode configurations, see Table 150, page 189 or Table 151, page 190.
11. Release the switch port from reset by clearing the reset bits in CLOCK_CFG.

It is not necessary to reset the SerDes macros.

## 6.3.4 Port Counters

The statistics collected in each port module provide monitoring of various events. This section describes how industry-standard Management Information Bases (MIBs) can be implemented using the counter set in this device. The following MIBs are considered:

- RMON statistics group (RFC 2819)
- IEEE 802.3-2005 Annex 30A counters
- SNMP interfaces group (RFC 2863)
- SNMP Ethernet-like group (RFC 3536)

### 6.3.4.1 RMON Statistics Group (RFC 2819)

The following table provides the mapping of RMON counters to port counters.

***Table 154 •*** **Mapping of RMON Counters to Port Counters**

| RMON Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| EtherStatsDropEvents | Rx | C_RX_CAT_DROP + C_DR_TAIL + sum of C_DR_YELLOW_PRIO_x + sum of C_DR_GREEN_PRIO_x, where x is 0 through 7. |
| EtherStatsOctets | Rx | C_RX_OCT |
| EtherStatsPkts | Rx | C_RX_SHORT + C_RX_FRAG + C_RX_JABBER + C_RX_LONG + C_RX_SZ_64 + C_RX_SZ_65_127 + C_RX_SZ_128_255 + C_RX_SZ_256_511 + C_RX_SZ_512_1023 + C_RX_SZ_1024_1526 + C_RX_SZ_JUMBO |
| EtherStatsBroadcastPkts | Rx | C_RX_BC |
| EtherStatsMulticastPkts | Rx | C_RX_MC |
| EtherStatsCRCAlignErrors | Rx | C_RX_CRC |
| EtherStatsUndersizePkts | Rx | C_RX_SHORT |
| EtherStatsOversizePkts | Rx | C_RX_LONG |
| EtherStatsFragments | Rx | C_RX_FRAG |
| EtherStatsJabbers | Rx | C_RX_JABBER |
| EtherStatsPkts64Octets | Rx | C_RX_SZ_64 |
| EtherStatsPkts65to127Octets | Rx | C_RX_SZ_65_127 |
| EtherStatsPkts128to255Octets | Rx | C_RX_SZ_128_255 |
| EtherStatsPkts256to511Octets | Rx | C_RX_SZ_256_511 |
| EtherStatsPkts512to1023Octets | Rx | C_RX_SZ_512_1023 |
| EtherStatsPkts1024to1518Octets | Rx | C_RX_SZ_1024_1526 |
| EtherStatsDropEvents | Tx | C_TX_DROP + C_TX_AGE |
| EtherStatsOctets | Tx | C_TX_OCT |
| EtherStatsPkts | Tx | C_TX_SZ_64 + C_TX_SZ_65_127 + C_TX_SZ_128_255 + C_TX_SZ_256_511 + C_TX_SZ_512_1023 + C_TX_SZ_1024_1526 + C_TX_SZ_JUMBO |
| EtherStatsBroadcastPkts | Tx | C_TX_BC |
| EtherStatsMulticastPkts | Tx | C_TX_MC |
| EtherStatsCollisions | Tx | C_TX_COL |
| EtherStatsPkts64Octets | Tx | C_TX_SZ_64 |
| EtherStatsPkts65to127Octets | Tx | C_TX_SZ_65_127 |
| EtherStatsPkts128to255Octets | Tx | C_TX_SZ_128_255 |
| EtherStatsPkts256to511Octets | Tx | C_TX_SZ_256_511 |
| EtherStatsPkts512to1023Octets | Tx | C_TX_SZ_512_1023 |
| EtherStatsPkts1024to1518Octets | Tx | C_TX_SZ_1024_1526 |

### 6.3.4.2 IEEE 802.3-2005 Annex 30A Counters

This section provides the mapping of IEEE 802.3-2005 Annex 30A counters to port counters. Only counter groups with supported counters are listed.

***Table 155 •*** **Mandatory Counters**

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| aFramesTransmittedOK | Tx | C_TX_SZ_64 + C_TX_SZ_65_127 + C_TX_SZ_128_255 + C_TX_SZ_256_511 + C_TX_SZ_512_1023 + C_TX_SZ_1024_1526 + C_TX_SZ_JUMBO |
| aSingleCollisionFrames | Tx | Does not apply |
| aMultipleCollisionFrames | Tx | Does not apply |
| aFramesReceivedOK | Rx | Sum of C_RX_GREEN_PRIO_x + C_RX_YELLOW_PRIO_x, where x is 0 through 7. |
| aFrameCheckSequenceErrors | Rx | Not available. C_RX_CRC is the sum of FCS and alignment errors. |
| aAlignmentErrors | Rx | Not available. C_RX_CRC is the sum of FCS and alignment errors. |

***Table 156 •*** **Optional Counters**

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| aMulticastFramesXmittedOK | Tx | C_TX_MC |
| aBroadcastFramesXmittedOK | Tx | C_TX_BC |
| aMulticastFramesReceivedOK | Rx | C_RX_MC |
| aBroadcastFramesReceivedOK | Rx | C_RX_BC |
| aInRangeLengthErrors | Rx | Not available |
| aOutOfRangeLengthField | Rx | Not available |
| aFrameTooLongErrors | Rx | C_RX_LONG |

***Table 157 •*** **Recommended MAC Control Counters**

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| aMACControlFramesTransmitted | Tx | Not available |
| aMACControlFramesReceived | Rx | C_RX_CONTROL |
| aUnsupportedOpcodesReceived | Rx | Not available |

***Table 158 •*** **Pause MAC Control Recommended Counters**

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| aPauseMACControlFramesTransmitted | Tx | C_TX_PAUSE. Transmitted pause frames in 10/100 Mbps full-duplex are not counted. |
| aPauseMACControlFramesReceived | Rx | C_RX_PAUSE |

### 6.3.4.3 SNMP Interfaces Group (RFC 2863)

The following table provides the mapping of SNMP interfaces group counters to port counters.

***Table 159 •*** **Mapping of SNMP Interfaces Group Counters to Port Counters**

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| IfInOctets | Rx | C_RX_OCT |
| IfInUcastPkts | Rx | C_RX_UC |
| IfInNUcastPkts | Rx | C_RX_BC + C_RX_MC |
| IfInBroadcast (RFC 1573) | Rx | C_RX_BC |
| IfInMulticast (RFC 1573) | Rx | C_RX_MC |
| IfInDiscards | Rx | C_DR_TAIL + C_RX_CAT_DROP |
| IfInErrors | Rx | C_RX_CRC + C_RX_SHORT + C_RX_FRAG + C_RX_JABBER + C_RX_LONG |
| IfInUnknownProtos | Rx | Always zero. |
| IfOutOctets | Tx | C_TX_OCT |
| IfOutUcastPkts | Tx | C_TX_UC |
| IfOutNUcastPkts | Tx | C_TX_BC + C_TX_MC |
| ifOutMulticast (RFC 1573) | Tx | C_TX_MC |
| ifOutBroadcast (RFC 1573) | Tx | C_TX_BC |
| IfOutDiscards | Tx | Always zero. |
| IfOutErrors | Tx | C_TX_DROP + C_TX_AGE |

### 6.3.4.4 SNMP Ethernet-Like Group (RFC 3536)

The following table provides the mapping of SNMP Ethernet-like group counters to port counters.

***Table 160 •*** **Mapping of SNMP Ethernet-Like Group Counters to Port Counters**

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| dot3StatsAlignmentErrors | Rx | Not available. C_RX_CRC is the sum of FCS and alignment errors. |
| dot3StatsFCSErrors | Rx | Not available. C_RX_CRC is the sum of FCS and alignment errors. |
| dot3StatsSingleCollisionFrames | Tx | Not available. |
| dot3StatsMultipleCollisionFrames | Tx | Not available. |
| dot3StatsSQETestErrors | Rx | Not applicable. |
| dot3StatsDeferredTransmissions | Tx | Not available. |
| dot3StatsLateCollisions | Tx | Not available. C_TX_DROP is the sum of Late collisions and Excessive collisions. |
| dot3StatsExcessiveCollisions | Tx | Not available. C_TX_DROP is the sum of Late collisions and Excessive collisions. |
| dot3StatsInternalMacTransmitErrors | Tx | Not applicable. Always 0. |
| dot3StatsCarrierSenseErrors | Tx | Not available. |
| dot3StatsFrameTooLongs | Rx | C_RX_LONG. |
| dot3StatsInternalMacReceiveErrors | Rx | Not applicable. Always 0. |

*Table 160 •* **Mapping of SNMP Ethernet-Like Group Counters to Port Counters *(continued)***

| Counter | Rx/Tx | Switch Core Implementation |
|---|---|---|
| dot3InPauseFrames | Rx | C_RX_PAUSE. |
| dot3OutPauseFrames | Tx | C_TX_PAUSE. Transmitted pause frames in 10/100 Mbps full-duplex are not counted. |

# 6.4 Layer-2 Switch

This section describes the Layer-2 switch features:

- Switching
- VLAN and GVRP
- Rapid and Multiple Spanning Tree
- Link aggregation
- Port-based access control
- Mirroring
- SNMP support

## 6.4.1 Basic Switching

Basic switching covers forwarding, address learning, and address aging.

### 6.4.1.1 Forwarding

The devices contain a Layer-2 switch and frames are forwarded using Layer-2 information only. Exceptions to this are possible using VCAP capabilities. For example, to provide source-specific IP multicast forwarding.

The switch is designed to comply with the IEEE Bridging standard in IEEE 802.1D and the IEEE VLAN standard in IEEE 802.1Q:

- Unicast frames are forwarded to a single destination port that corresponds to the DMAC.
- Multicast frames are forwarded to multiple ports determined by the DMAC multicast group. The CPU configures multicast groups in the MAC table and the port group identifier (PGID) table. A multicast group can span across any set of ports.
- Broadcast frames (DMAC = FF-FF-FF-FF-FF-FF) are, by default, flooded to all ports except the ingress port. Also, in compliance with the standard, a unicast or multicast frame with unknown DMAC is flooded to all ports except the ingress port. It is possible to configure flood masks to restrict the flooding of frames. There are separate flood masks for the following frame types:

  Unicast (ANA::FLOODING.FLD_UNICAST)
  Layer 2 multicast (ANA::FLOODING.FLD_MULTICAST)
  Layer 2 broadcast (ANA::FLOODING.FLD_BROADCAST)
  IPv4 multicast data (ANA::FLOODING_IPMC.FLD_MC4_DATA)
  IPv4 multicast control (ANA::FLOODING_IPMC.FLD_MC4_CTRL)
  IPv6 multicast data (ANA::FLOODING_IPMC.FLD_MC6_DATA)
  IPv6 multicast control (ANA::FLOODING_IPMC.FLD_MC6_CTRL)

For frames with a known destination MAC address, the destination mask comes from an entry in the port group identifier table (ANA::PGID). The PGID table contains 107 entries (entry 0 through 106), where entry 0 through 63 are used for destination masks. The remaining PGID entries are used for other parts of the forwarding and are described below.

The following table shows the PGID table organization.

*Table 161 •* **Port Group Identifier Table Organization**

| Entry Type | Number |
|---|---|
| Unicast entries | 0 – 26 (including CPU) |
| Multicast entries | 27 – 63 |

*Table 161 •* **Port Group Identifier Table Organization**

| Entry Type | Number |
|---|---|
| Aggregation Masks | 64 – 79 |
| Source Masks | 80 – 106 |

The unicast entries contains only the port number corresponding to the entry number.

Destination masks for multicast groups must be manually entered through the CPU into the destination masks table. IPv4 and IPv6 multicast entries can also be entered using direct encoding in the MAC table, where the destination masks table is not used. For information about forwarding and configuring destination masks, see MAC Table, page 85.

The aggregation masks ensures that a frame is forwarded to exactly one member of an aggregation group.

For all forwarding decisions, a source mask prevents frames from being sent back to the ingress port. The source mask removes the ingress port from the destination mask.

All ports are enabled for receiving frames by default. This can be disabled by clearing ANA:PORT:PORT_CFG.RECV_ENA.

### 6.4.1.2 Address Learning

The learning process minimizes the flooding of frames. A frame's source MAC address is learned together with its VID. Each entry in the MAC table is uniquely identified by a (MAC,VID) pair. In the forwarding process, a frame's (DMAC,VID) pair is used as the key for the MAC table lookup.

The learning of unknown SMAC addresses can be either hardware-based or CPU-based. The following list shows the available learn schemes, which can be configured per port:

- **Hardware-based learning** autonomously adds entries to the MAC table without interaction from the CPU. Use the following configuration:
  ANA:PORT:PORT_CFG.LEARN_ENA = 1
  ANA:PORT:PORT_CFG.LEARNCPU = 0
  ANA:PORT:PORT_CFG.LEARNDROP = 0
  ANA:PORT:PORT_CFG.LEARNAUTO = 1
- **CPU-based learning** copies frames with unknown SMACs, or when the SMAC appears on a different port, to the CPU. These frames are forwarded as usual. Use the following configuration.
  ANA:PORT:PORT_CFG.LEARN_ENA = 1
  ANA:PORT:PORT_CFG.LEARNCPU = 1
  ANA:PORT:PORT_CFG.LEARNDROP = 0
  ANA:PORT:PORT_CFG.LEARNAUTO = 0
- **Secure CPU-based learning** is similar to CPU-based learning, except that it allows the CPU to verify the SMAC addresses before both learning and forwarding. Secure CPU-based learning redirects frames with unknown SMACs, or when the SMAC appears on a different port, to the CPU. These frames are not forwarded by hardware. Use the following configuration.
  ANA::PORT_CFG.LEARN_ENA = 1
  ANA::PORT_CFG.LEARNCPU = 1
  ANA::PORT_CFG.LEARNDROP = 1
  ANA::PORT_CFG.LEARNAUTO = 0
- **No learning** where all learn frames are discarded. Frames with known SMAC in the MAC table are forwarded by hardware. Use the following configuration.
  ANA:PORT:PORT_CFG.LEARN_ENA = 1
  ANA:PORT:PORT_CFG.LEARNCPU = 0
  ANA:PORT:PORT_CFG.LEARNDROP = 1
  ANA:PORT:PORT_CFG.LEARNAUTO = 0

Frames forwarded to the CPU for learning can be extracted from the CPU extraction queue configured in ANA:PORT:CPUQ_CFG.CPUQ_LRN.

During CPU-based learning, the rate of frames subject to learning being copied or redirected to the CPU can be controlled with the learn storm policer (ANA::STORMLIMIT_CFG[3]). This policer puts a limit on

the number of frames per second that are subject to learning being copied or redirected to the CPU. The learn frames storm policer can help prevent a CPU from being overloaded when performing CPU based learning.

#### 6.4.1.3 MAC Table Address Aging

To keep the MAC table updated, an aging scan is conducted to remove entries that were not recently accessed. This ensures that stations that have moved to a new location are not permanently prevented from receiving frames in their new location. It also frees up MAC table entries occupied by obsolete stations to give room for new stations.

In IEEE 802.1D, the recommended period for aging-out entries in the MAC address table is 300 seconds per entry. The device aging implementation checks for the aging-out of all the entries in the table. The first age scan sets the age bit for every entry in the table. The second age scan removes entries where the age bit has not been cleared since the first age scan. An entry's age bit is cleared when a received frame's (SMAC, VID) matches an entry's (MAC, VID); that is, the station is active and transmits frames. To ensure that 300 seconds is the longest an entry can reside not accessed (and unchanged) in the table, the maximum time between age scans is 150 seconds.

The device can conduct age scans in two ways:

- Automatic age scans
- CPU initiated age scans

When using automatic aging, the time between age scans is set in the ANA::AUTOAGE register in steps of 1 second, in the range from 1 second to 12 days.

When using CPU-initiated aging, the CPU implements the timing between age scans. A scan is initiated by sending an aging command to the MAC address table (ANA::MACACCESS. MAC_TABLE_CMD).

The CPU-controlled age scan process can conveniently be used to flush the entire MAC table by conducting two age scans, one immediately after the other.

Flushing selective MAC table entries is also possible. Incidents that require MAC table flushing are:

- Reconfiguration of Spanning Tree protocol port states, which may cause station moves to occur.
- If there is a link failure notification (identified by a PHY layer device), flush the MAC table on the specific port where the link failed.

To deal with these incidents, the age scan process is configurable to run only for entries learned on a specified port or for a specified VLAN (ANA:: ANAGEFIL.VID_VAL). The filters can also be combined to do aging on entries that match both the specific port and the specific VLAN.

Single entries can be flushed from the MAC table by sending the FORGET command to the MAC address table.

### 6.4.2 Standard VLAN Operation

This section provides information about configuring and operating the devices as a standard VLAN-aware switch. For more information about using the switch as a Q-in-Q enabled provider bridge, see Provider Bridges and Q-in-Q Operation, page 200. For information about the use of private VLANs and asymmetric VLANs, see Private VLANs, page 205 and Asymmetric VLANs, page 209.

The following table lists the port module registers for standard VLAN operation.

***Table 162 •*** **Port Module Registers for Standard VLAN Operation**

| Register/Register Field | Description | Replication |
|---|---|---|
| MAC_TAGS_CFG | Allows tagged frames to be 4 bytes longer than the length configured in MAC_MAXLEN_CFG. | Per port |

The following table lists the analyzer configurations and status bits for standard VLAN operation.

***Table 163 •*** **Analyzer Registers for Standard VLAN Operation**

| Register/Register Field | Description | Replication |
|---|---|---|
| DROP_CFG.DROP_UNTAGGED_ENA | Discard untagged frames. | Per port |
| DROP_CFG.DROP_C_TAGGED_ENA | Discard VLAN tagged frames. | Per port |
| DROP_CFG.DROP_PRIO_C_TAGGED_ENA | Discard priority tagged frames. | Per port |
| VLAN_CFG.VLAN_AWARE_ENA | Use incoming VLAN tags in VLAN classification. | Per port |
| VLAN_CFG.VLAN_POP_CNT | Remove VLAN tags from frames in the rewriter. | Per port |
| VLAN_CFG.VLAN_DEI<br>VLAN_CFG.VLAN_PCP<br>VLAN_CFG.VLAN_VID | Ingress port VLAN configuration. | Per port |
| VLANMASK | Per-port VLAN ingress filtering enable. | None |
| ANEVENTS.VLAN_DISCARD | A sticky bit indicating that a frame was dropped due to lack of VLAN membership of source port. | None |
| ADVLEARN.VLAN_CHK | Disable learning for frames discarded due to source port VLAN membership check. | None |
| VLANACCESS | VLAN table command. For indirect access to configuration of the 4096 VLANs. | None |
| VLANTIDX | VLAN table index. For indirect access to configuration of the 4096 VLANs. | None |
| AGENCTRL.FID_MASK | Enable shared VLAN learning. | None |
| CPU_FWD_GARP_CFG | Enable capture of frames with reserved GARP DMAC addresses, including GVRP for VLAN registration. Per-address configuration. | Per port |
| CPUQ_8021_CFG.CPUQ_GARP_VAL | CPU queue for captured GARP frames. | Per GARP address |

The following table lists the rewriter registers for standard VLAN operation.

***Table 164 •*** **Rewriter Registers for Standard VLAN Operation**

| Register/Register Field | Description | Replication |
|---|---|---|
| TAG_CFG | Egress VLAN tagging configuration | Per port |
| PORT_VLAN_CFG | Egress port VLAN configuration | Per port |

In a VLAN-aware switch, each port is a member of one or more virtual LANs. Each incoming frame must be assigned a VLAN membership and forwarded according to the assigned VID. The following information draws on the definitions and principles of operations in IEEE 802.1Q. Note that the switch supports more features than mentioned in the following section, which only describes the basic requirements for a VLAN aware switch.

Standard VLAN operation is configured individually per switch port using the following configuration:

- MAC_TAGS_CFG.VLAN_AWR_ENA = 1
  MAC_TAGS_CFG.VLAN_LEN_AWR_ENA = 1
- VLAN_CFG.VLAN_AWARE_ENA = 1,
  VLAN_CFG.VLAN_POP_CNT = 1.

Each switch port has an Acceptable Frame Type parameter, which is set to Admit Only VLAN tagged frames or Admit All Frames:

- Admit Only VLAN-tagged frames:
  DROP_CFG.DROP_UNTAGGED_ENA = 1,
  DROP_CFG.DROP_PRIO_C_TAGGED_ENA = 1,
  DROP_CFG.DROP_C_TAGGED = 0.
- Admit All Frames:
  DROP_CFG.DROP_UNTAGGED_ENA = 0,
  DROP_CFG.DROP_PRIO_C_TAGGED_ENA = 0,
  DROP_CFG.DROP_C_TAGGED = 0.

Frames that are not discarded are subject to the VLAN classification. Untagged and priority-tagged frames are classified to a Port VLAN Identifier (PVID). The PVID is configured per port in VLAN_CFG.VLAN_VID. Tagged frames are classified to the VID given in the frame's tag. For more information about VLAN classification, see VLAN Classification, page 55.

#### 6.4.2.1 Forwarding

Forwarding is always based on the combination of the classified VID and the destination MAC address. By default, all switch ports are members of all VLANs. This can be changed in VLANACCESS and VLANTIDX where port masks per VLAN are set up.

#### 6.4.2.2 Ingress Filtering

VLAN ingress filtering can be enabled per switch port with the register VLANMASK and per router port with MACx_CFG.INGRESS_CHK.

The filter checks for all incoming frames to determine if the ingress port is a member of the VLAN to which the frame is classified. If the port is not a member, the frame is discarded. Whenever a frame is discarded due to lack of VLAN membership, the ANEVENTS.VLAN_DISCARD sticky bit is set. To ensure that VLAN ingress filtered frames are not learned, ADVLEARN.VLAN_CHK must be set.

#### 6.4.2.3 GARP VLAN Registration Protocol (GVRP)

GARP VLAN Registration Protocol (GVRP) is used to propagate VLAN configurations between bridges. On a GVRP-enabled switch, all GVRP frames must be redirected to the CPU for further processing. The GVRP frames use a reserved GARP MAC address (01-80-C2-00-00-21) and can be redirected to the CPU by setting bit 1 in the analyzer register CPU_FWD_GARP_CFG.

#### 6.4.2.4 Shared VLAN Learning

The devices can be configured for either Independent VLAN learning or Shared VLAN learning. Independent VLAN learning is the default.

Shared VLAN learning, where multiple VLANs map to the same filtering database, is enabled through Filter Identifiers (FIDs). Basically, this means that learning is unique for a (MAC, FID) set and that a learned MAC address is learned for all VIDs that map to the FID. Shared VLAN learning is enabled in AGENCTRL.FID_MASK.

The 12-bit FID mask sets which bits in the VID are indifferent to the learning. For example, if the least significant two bits are set in the FID mask, the following VID sets are sharing learning, where X and Y are any hexadecimal digits:

- VID set 1: 0xXY0, 0xXY1, 0xXY2, 0xXY3
- VID set 2: 0xXY4, 0xXY5, 0xXY6, 0xXY7
- VID set 3: 0xXY8, 0xXY9, 0xXYA, 0xXYB
- VID set 4: 0xXYC, 0xXYD, 0xXYE, 0xXYF

#### 6.4.2.5 Untagging

An untagged set can be configured for each egress port, which defines the VIDs for which frames are transmitted untagged. The untagged set can consist of zero, one, or all VIDs. For all VIDs not in the untagged set, frames are transmitted tagged. The available configurations are:

- The untagged set is empty:
  TAG_CFG.TAG_CFG = 3.

- The untagged set consists of all VIDs:
  TAG_CFG.TAG_CFG = 0.
- The untagged set consists of one VID <VID>:
  TAG_CFG.TAG_CFG = 1.
  PORT_VLAN_CFG.PORT_VID = <VID>.

Optionally, frames received as priority-tagged frames (VID = 0) can also be transmitted as untagged (TAG_CFG.TAG_CFG=2).

#### 6.4.2.5.1 Port-Based VLAN Example

**Situation:**

Ports 0 and 1 are isolated from ports 2 and 3 using port-based VLANs. Ports 0 and 1 are assigned port VID 1 and ports 2 and 3 port VID 2. All frames in the network are untagged.

**Resolution:**

```
# Port module configuration of ports 0 – 1.
# Configure the ports to always use the port VID.
VLAN_CFG.VLAN_AWARE_ENA = 0
# Allow only untagged frames.
DROP_CFG.DROP_UNTAGGED_ENA = 0
DROP_CFG.DROP_PRIO_C_TAGGED = 1
DROP_CFG.DROP_C_TAGGED = 1
# Configure the port VID to 1.
VLAN_CFG.VLAN_VID = 1

# Port module configuration of ports 2 – 3.
# Same as for ports 0-1, except that the port VID is set to 2.
VLAN_CFG.VLAN_VID = 2
# Analyzer configuration.
# Configure VLAN 1 to contain ports 0-1.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x03
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 2 to contain ports 2-3.
VLANTIDX.INDEX = 2
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x0C
VLANACCESS.VLAN_TBL_CMD = 2
```

### 6.4.3 Provider Bridges and Q-in-Q Operation

The following table lists the port module configurations for provider bridge VLAN operation.

***Table 165 •*** **Port Module Configurations for Provider Bridge VLAN Operation**

| Register/Register Field | Description | Replication |
|---|---|---|
| MAC_TAGS_CFG | Allow single tagged frames to be 4 bytes longer and double-tagged frames to be 8 bytes longer than the length configured in MAC_MAXLEN_CFG. | Per port |

The following table lists the port module configurations for provider bridge VLAN operation.

***Table 166 •*** **System Configurations for Provider Bridge VLAN Operation**

| Register/Register Field | Description | Replication |
|---|---|---|
| VLAN_ETYPE_CFG.VLAN_S_TAG_ETYPE_VAL | TPID for S-tagged frames. EtherType 0x88A8 and the configurable value VLAN_ETYPE_CFG.VLAN_S_TAG_ETYPE_VAL are identified as the S-tag identifier. | Per port |

The following table lists the analyzer configurations for provider bridge VLAN operation.

***Table 167 •*** **Analyzer Configurations for Provider Bridge VLAN Operation**

| Register/Register Field | Description | Replication |
|---|---|---|
| DROP_CFG.DROP_UNTAGGED_ENA | Discard untagged frames. | Per port |
| DROP_CFG.DROP_S_TAGGED_ENA | Discard VLAN S-tagged frames. | Per port |
| DROP_CFG.DROP_PRIO_S_TAGGED_ENA | Discard priority S-tagged frames. | Per port |
| VLAN_CFG.VLAN_AWARE_ENA | Use incoming VLAN tags in VLAN classification. | Per port |
| VLAN_CFG.VLAN_POP_CNT | Remove VLAN tags from frames in the rewriter. | Per port |
| VLAN_CFG.VLAN_TAG_TYPE | Tag type for untagged frames (Customer tag or service tag). | Per port |
| VLAN_CFG.VLAN_INNER_TAG_ENA | Use inner tag for VLAN classification instead of outer tag. | Per port |
| VLAN_CFG.VLAN_DEI<br>VLAN_CFG.VLAN_PCP<br>VLAN_CFG.VLAN_VID | Ingress port VLAN configuration. | Per port |
| VLANACCESS | VLAN table command. For indirect access to configuration of the 4096 VLANs. | None |
| VLANTIDX | VLAN table index. For indirect access to configuration of the 4096 VLANs. | None |

The devices support the standard provider bridge features in IEEE 802.1ad (Provider Bridges). The features related to provider bridges are:

- Support for multiple tag headers (EtherTypes 0x8100, 0x88A8, and a programmable value are recognized as tag header EtherTypes)
- Pushing and popping of up to two VLAN tags
- Selective VLAN classification using either inner or outer VLAN tag
- Translating VLAN tag headers at ingress and/or at egress (using the IS1 and ES0 TCAMs)
- Enabling or disabling learning per VLAN

The following section discusses briefly how to configure these different features in the switch.

The devices support multiple VLAN tags. They can be used in MAN applications as a provider bridge, aggregating traffic from numerous independent customer LANs into the MAN space. One of the purposes of the provider bridge is to recognize and use VLAN tags so that the VLANs in the MAN space can be used independent of the customers' VLANs. This is accomplished by adding a VLAN tag with a

MAN-related VID for frames entering the MAN. When leaving the MAN, the tag is stripped, and the original VLAN tag with the customer-related VID is again available. This provides a tunneling mechanism to connect remote customer VLANs through a common MAN space without interfering with the VLAN tags. All tags use EtherType 0x8100 for customer tags and EtherType 0x88A8, or a programmable value, for service provider tags.

If a given service VLAN only has two member ports on the switch, the learning can be disabled for the particular VLAN (VLANTIDX.VLAN_LEARN_DISABLE) and can rely on flooding as the forwarding mechanism between the two ports. This way, the MAC table requirements are reduced.

#### 6.4.3.0.1 MAN Access Switch Example

**Situation:**

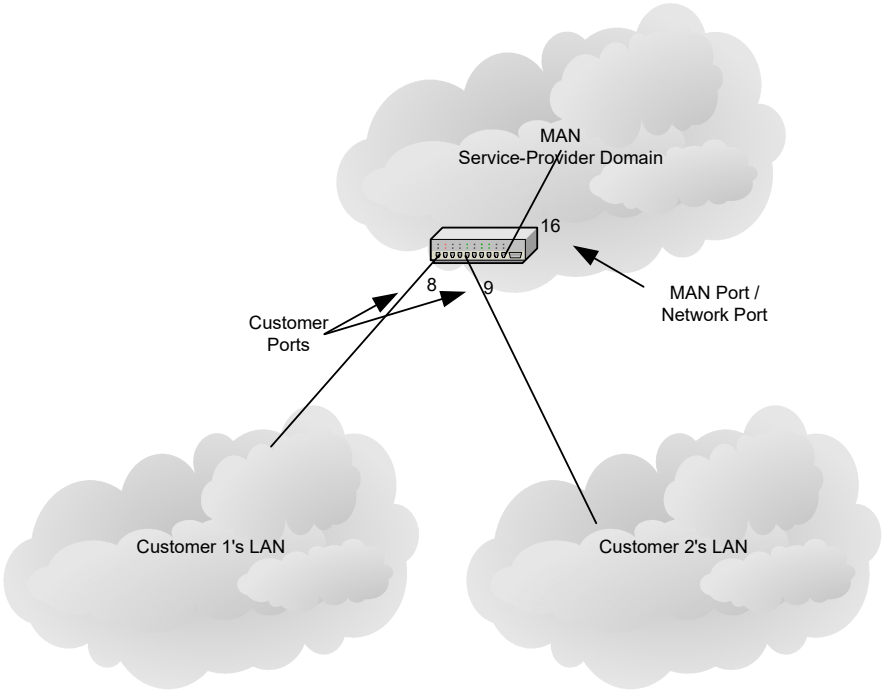
The following is an example of setting up the device as a MAN access switch with these requirements:

- Customer ports are aggregated into a network port for tunneling through the MAN to access remote VLANs.
- Local switching between ports of the different customers must be eliminated.
- Frames must be label-switched from network port to correct customer port without need for MAC address learning.

***Figure 69 •*** **MAN Access Switch Setup**

Frames in This Segment

| Service Provider Tag (Outer Tag) | | Customer Tag (Inner Tag) | | |
|---|---|---|---|---|
| EtherType | VID | EtherType | VID | Description |
| 0x88A8 | 1 | 0x8100 | 1 | Frames to/from customer 1's VLAN 1 |
| 0x88A8 | 1 | 0x8100 | 118 | Frames to/from customer 1's VLAN 118 |
| 0x88A8 | 1 | 0x8100 | 0 | Priority-tagged frames to/from customer 1 |
| 0x88A8 | 2 | 0x8100 | 1 | Frames to/from customer 2's VLAN 1 |
| 0x88A8 | 2 | 0x8100 | 4 | Frames to/from customer 2's VLAN 4 |
| 0x88A8 | 2 | N/A | N/A | Untagged frames to/from customer 2 |




Frames in This Segment

| Customer Tag | | |
|---|---|---|
| EtherType | VID | Description |
| 0x8100 | 1 | Frames in Customer 1's VLAN 1 |
| 0x8100 | 118 | Frames in Customer 1's VLAN 118 |
| 0x8100 | 0 | Customer 1's Priority-Tagged Frames |

Frames in This Segment

| Customer Tag | | |
|---|---|---|
| EtherType | VID | Description |
| 0x8100 | 1 | Frames in Customer 2's VLAN 1 |
| 0x8100 | 4 | Frames in Customer 2's VLAN 4 |
| N/A | N/A | Customer 2's Untagged Frames |

This example is typically accomplished by letting each customer port have a unique port VID (PVID), which is used in the outer VLAN tag (the service provider tag). In the MAN, the VID directly indicates the customer port from which the frame is received or the customer port to which the frame is going.

A customer port is VLAN-unaware and classifies to a port-based VLAN. In the egress direction of the customer port, frames are transmitted untagged, which facilitates the stripping of the outer tag. That is, the provider tag is stripped, but the customer tag is kept. The port must allow frames with a maximum size of 1522 bytes.

**Resolution:**

```
# Configuration of customer 1’s port (port 8).
# Allow for a single VLAN tag in the length check and set the maximum length
without VLAN
# tag to 1518 bytes.
MAC_TAGS_CFG.VLAN_LEN_AWR_ENA = 1
MAC_TAGS_CFG.VLAN_AWAR_ENA = 1
MAC_MAXLEN_CFG.MAX_LEN = 1518
# Configure the port to leave any incoming tags in the frame and to ignore any
# incoming VLAN tags in the VLAN classification. The port VID is always used
in the
# VLAN classification.
VLAN_CFG.VLAN_POP_CNT = 0
VLAN_CFG.VLAN_AWARE_ENA = 0
# Allow both C-tagged and untagged frames coming in to the device to also
support customer traffic not using VLANs to be carried across the MAN.
DROP_CFG.DROP_UNTAGGED_ENA = 0
DROP_CFG.DROP_C_TAGGED = 0
DROP_CFG.DROP_PRIO_C_TAGGED = 0
DROP_CFG.DROP_S_TAGGED = 1
DROP_CFG.DROP_PRIO_S_TAGGED = 1
# Use service provider tagging when frames from this port exit the switch.
# (EtherType 0x88A8).
VLAN_CFG.VLANTAG_TYPE = 1
# Configure the port VID to 1.
VLAN_CFG.VLAN_VID = 1
# Configure the egress side of the port to not insert tags.
# (The service provider tags are stripped in the ingress side of the MAN port).
TAG_CFG.TAG_CFG = 0
# Configuration of customer 2's port (port 9).
# Same as for customer 1’s port (port 8), except that the port VID is set to 2.
VLAN_CFG.VLAN_VID = 2
# Configuration of the network port (port 16).
# MAN traffic in transit between network ports is supported by configuring all
network
# ports as follows:
# Allow for two VLAN tags in the length check and set the max length without
# VLAN tags to 1518 bytes.
MAC_TAGS_CFG.VLAN_LEN_AWR_ENA = 1
MAC_TAGS_CFG.VLAN_AWAR_ENA = 1
MAC_TAGS_CFG.PB_ENA =1
MAC_MAXLEN_CFG.MAX_LEN = 1518
# Configure the port to use incoming VLAN tags in the VLAN classification,
# and to remove the first (outer) VLAN tag (the service tag) from incoming
frames.
VLAN_CFG.VLAN_POP_CNT = 1
VLAN_CFG.VLAN_AWARE_ENA = 1
# Allow only S-tagged frames.
DROP_CFG.DROP_UNTAGGED_ENA = 1
DROP_CFG.DROP_C_TAGGED = 1
DROP_CFG.DROP_PRIO_C_TAGGED = 1
DROP_CFG.DROP_S_TAGGED = 0
DROP_CFG.DROP_PRIO_S_TAGGED = 0
# The tag type is unused on the network port
VLAN_CFG.VLANTAG_TYPE = 0
# Configure the egress side of the port to insert tags.
TAG_CFG.TAG_CFG = 1
# Common configuration in the analyzer.
```

```
# Configure VLAN 1 to contain customer 1’s port (port 8) and the network port
# (port 16). Disable learning in VLAN 1. Ingress filtering is don’t care for
port
# based VLANs.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 0 (don’t care, for this example)
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 1
VLANTIDX.VLAN_SRC_CHK = 0 (don’t care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x00010100
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 2 to contain customer 2's port (port 9) and the network port
# (port 16). Disable learning in VLAN 2. Ingress filtering is don’t-care for
port
# based VLANs.
VLANTIDX.INDEX = 2
VLANTIDX.VLAN_PRIV_VLAN = 0 (don’t care, for this example)
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 1
VLANTIDX.VLAN_SRC_CHK = 0 (don’t care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x00010200
VLANACCESS.VLAN_TBL_CMD = 2
```

### 6.4.4 Private VLANs

The following table lists the analyzer configuration registers for private VLAN support.

***Table 168 •*** **Private VLAN Configuration Registers**

| Register | Description | Replication |
|---|---|---|
| VLANACCESS | VLAN table command. For indirect access to configuration of the 4096 VLANs. | None |
| VLANTIDX | VLAN table index. For indirect access to configuration of the 4096 VLANs. | None |
| ISOLATED_PORTS | VLAN port mask indicating isolated ports in private VLANs. | None |
| COMMUNITY_PORTS | VLAN port mask indicating community ports in private VLANs. | None |

When a VLAN is configured to be a private VLAN, communication between ports within that VLAN can be prevented. Two application examples are:

- Customers connected to an ISP can be members of the same VLAN, but they are not allowed to communicate with each other within that VLAN.
- Servers in a farm of web servers in a Demilitarized Zone (DMZ) are allowed to communicate with the outside world and with database servers on the inside segment, but are not allowed to communicate with each other

For private VLANs to be applied, the switch must first be configured for standard VLAN operation. For more information, see Standard VLAN Operation, page 197. When this is in place, one or more of the configured VLANs can be configured as private VLANs. Ports in a private VLAN fall into one of three groups:

- Promiscuous ports
  Ports from which traffic can be forwarded to all ports in the private VLAN

  Ports that can receive traffic from all ports in the private VLAN
- Community Ports
  Ports from which traffic can only be forwarded to community and promiscuous ports in the private

VLAN

Ports that can receive traffic from only community and promiscuous ports in the private VLAN

- Isolated ports
  Ports from which traffic can only be forwarded to promiscuous ports in the private VLAN

  Ports that can receive traffic from only promiscuous ports in the private VLAN

The configuration of promiscuous, community, and isolated ports applies to all private VLANs.

The forwarding of frames classified to a private VLAN happens:

- When traffic comes in on a promiscuous port in a private VLAN, the VLAN mask from the VLAN table is applied.
- When traffic comes in on a community port, the ISOLATED_PORT mask is applied in addition to the VLAN mask from the VLAN table.
- When traffic comes in on an isolated port, the ISOLATED_PORT mask and the COMMUNITY_PORT mask are applied in addition to the VLAN mask from the VLAN table.

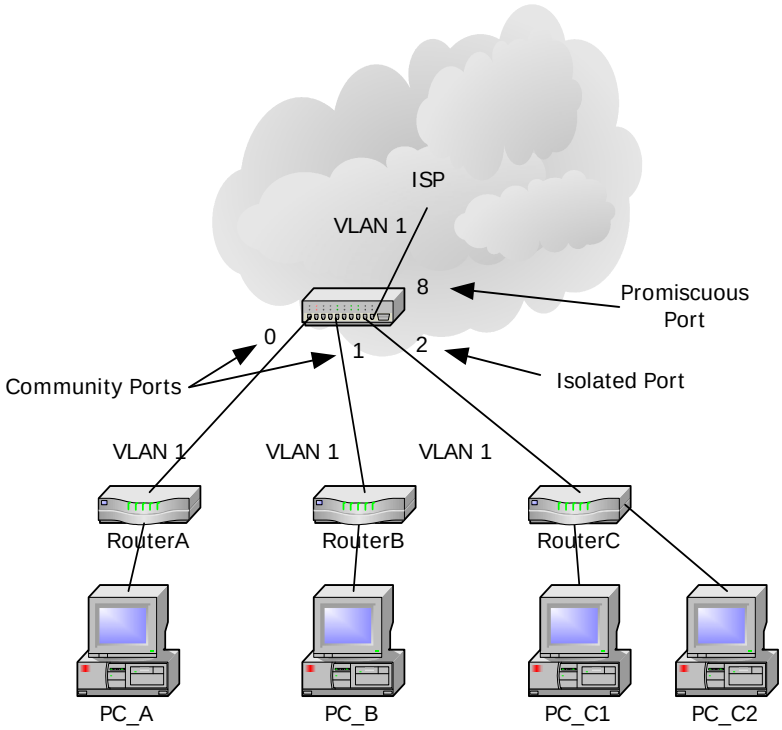
#### 6.4.4.0.1 ISP Example

**Situation:**

Customers A, B, and C are connected to the same switch at the ISP. Customers A and B are allowed to communicate with each other, as well as the ISP. Customer C can only communicate with the ISP. VLAN 1 is the private VLAN that isolates Customers A, B from C. Traffic on VLAN 1 coming in from the ISP (port 8) uses the VLAN mask in the VLAN table. Traffic on VLAN 1 from customer A or B has the ISOLATED_PORTS mask applied in addition to the mask from the VLAN table, with the result that traffic from customer A and B is not forwarded to customers C. Traffic on VLAN 1 from customer C has the ISOLATED_PORTS mask and the COMMUNITY_PORTS mask applied in addition to the mask from the VLAN table, with the result that traffic from customer C is not forwarded to customers A and B.

The following illustration shows the desired setup.

***Figure 70 •*** **ISP Example for Private VLAN**




VLAN Table

| VID | Ports in Mask | Private |
|---|---|---|
| ⋮ | ⋮ | ⋮ |
| 1 | 0, 1, 2, 8 | 1 |
| ⋮ | ⋮ | ⋮ |

ISOLATED_PORTS
(Isolated ports set to 0)

| Bit | Value |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 0 |
| ⋮ | ⋮ |
| 8 | 1 |
| ⋮ | ⋮ |

COMMUNITY_PORTS
(Community ports set to 0)

| Bit | Value |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 1 |
| ⋮ | ⋮ |
| 8 | 1 |
| ⋮ | ⋮ |

**Resolution:**

```
# It is assumed that Port VID and tag handling for VLAN 1 is already
# configured according to the description in Standard VLAN Operation.
# Configure VLAN 1 as a private VLAN in the VLAN table by performing these
steps:
# - Point to VLAN 1.
# - Set it as private.
# - Disable mirroring of the VLAN (not important for the example).
# - Enable learning within the VLAN (not important for the example).
# - Disable source check within the VLAN (not important for the example).
# - Include ports 0, 1, 2, and 8 in the VLAN mask.
# Insert the entry into the VLAN table.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 1
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0 (don’t care, for this example)
VLANTIDX.VLAN_SRC_CHK = 0 (don't care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x00000107
VLANACCESS.VLAN_TBL_CMD = 2
```

```
# Configure the private VLAN mask so that port 8 is a promiscuous
# port, ports 0 and 1 are community ports, and port 2 is an isolated port.
ISOLATED_PORTS.ISOL_PORTS = 0x00000103
COMMUNITY_PORTS.COMM_PORTS = 0x00000104
```

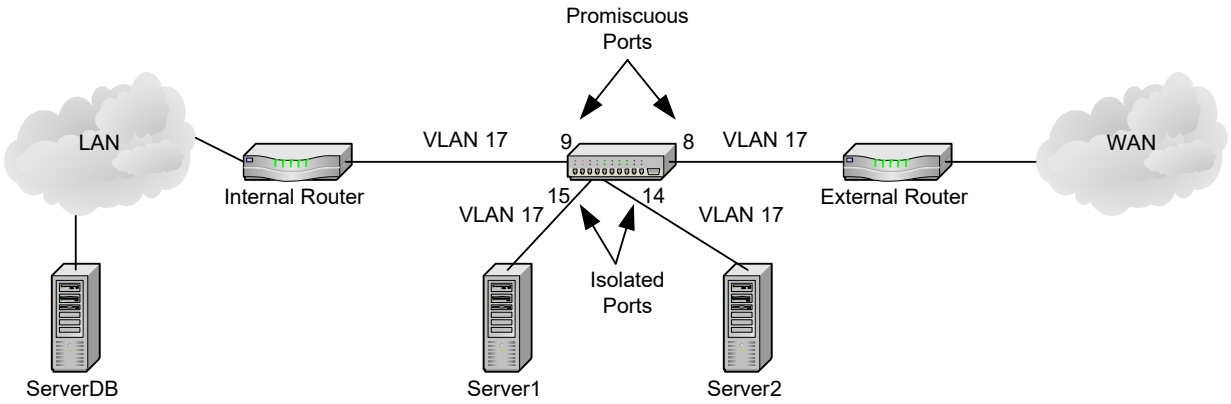
### 6.4.4.0.2 DMZ Example

**Situation:**

VLAN 17 is a private VLAN that isolates Server1 and Server2. Traffic on VLAN 17 coming from the internal or the external router (ports 8 and 9) uses the VLAN mask in the VLAN table. Traffic on VLAN 17 from Server1 and Server2 (ports 14 and 15) has the ISOLATED_PORTS applied in addition to the mask from the VLAN table, with the result that traffic from Server1 is not forwarded to Server2 and visa versa.

The following illustration shows the desired setup.

***Figure 71 •*** **DMZ Example for Private VLAN**




VLAN Table

| VID | Ports in Mask | Private |
|---|---|---|
| ⋮ | ⋮ | ⋮ |
| 17 | 8, 9, 14, 15 | 1 |
| ⋮ | ⋮ | ⋮ |

ISOLATED_PORTS
(Promiscuous Ports Set to 1)

| Bit | Value |
|---|---|
| ⋮ | ⋮ |
| 8 | 1 |
| 9 | 1 |
| ⋮ | ⋮ |
| 14 | 0 |
| 15 | 0 |
| ⋮ | ⋮ |

**Resolution:**

```
# It is assumed that Port VID and tag handling for VLAN 17 is already
# configured according to the description in Standard VLAN Operation.
# Configure VLAN 17 as a private VLAN in the VLAN table by performing these
steps:
# - Point to VLAN 17.
# - Set it as private.
# - Disable mirroring of the VLAN (not important for the example).
# - Enable learning within the VLAN (not important for the example).
# - Disable source check within the VLAN (not important for the example).
# - Include ports 8, 9, 14, and 15 in the VLAN mask.
# - Insert the entry into the VLAN table.
VLANTIDX.INDEX = 17
VLANTIDX.VLAN_PRIV_VLAN = 1
```

```
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0 (don’t care, for this example)
VLANTIDX.VLAN_SRC_CHK = 0 (don’t care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x0000C300
VLANACCESS.VLAN_TBL_CMD = 2
# Configure the private VLAN mask so that ports 8 and 9 are promiscuous
# ports.
ISOLATED_PORTS.ISOL_PORTS = 0x00000300
```

## 6.4.5 Asymmetric VLANs

Asymmetric VLANs use the same configuration registers as for standard VLAN operation. For more information about standard VLAN operation, see Standard VLAN Operation, page 197.

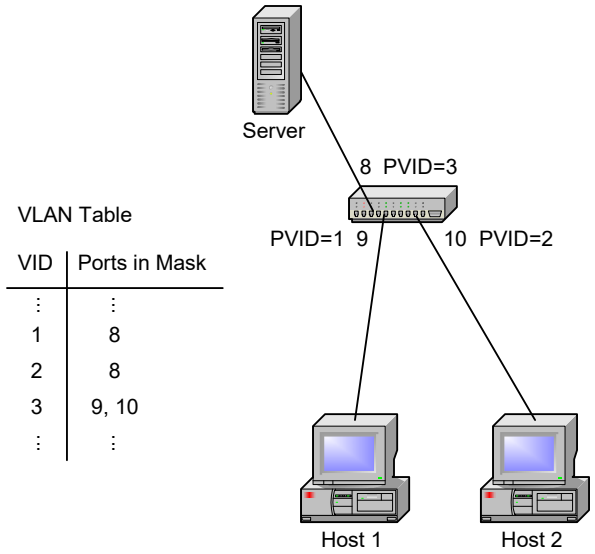
Asymmetric VLANs can be used to prevent communication between hosts in a network. This behavior is similar to what can be obtained by using private VLANs. For more information, seePrivate VLANs, page 205.

**Situation:**

A server and two hosts are connected to a switch. Communication between the hosts and the server should be allowed, but the hosts are not allowed to communicate directly. All traffic between the server and the hosts is untagged. Host 1 is connected to port 9, host 2 to port 10, and the server to port 8.

The host-1 port gets port VID 1 and the host-2 port gets port VID 2. The server port is a member of both VLANs 1 and 2. The server port gets port VID 3, and the two host ports are members of VLAN 3, as shown in the following illustration.

***Figure 72 •*** **Asymmetric VLANs**




**Resolution:**

```
# Analyzer configurations common for ports 8, 9, and 10.
# Allow only untagged frames.
DROP_CFG.DROP_UNTAGGED_ENA = 0
DROP_CFG.DROP_C_TAGGED_ENA = 1
DROP_CFG.DROP_PRIO_C_TAGGED_ENA = 1
# As tagged frames are dropped all frames are classified to the port VID.
VLAN_CFG.VLAN_AWARE_ENA = 0 (don’t care, for this example)
# Configure the egress side of the port to not insert tags.
TAG_CFG.TAG_CFG = 0
```

```
# Analyzer configuration specific for port 8. Set the port VID to 3.
VLAN_CFG.VLAN_VID = 3
VLAN_CFG.VLAN_DEI = 0 (don’t care, for this example)
# Analyzer configuration specific for port 9. Set the port VID to 1.
VLAN_CFG.VLAN_VID = 1
VLAN_CFG.VLAN_DEI = 0 (don’t care, for this example)


# Analyzer configuration specific for port 10. Set the port VID to 2.
VLAN_CFG.VLAN_VID = 2
VLAN_CFG.VLAN_DEI = 0 (don’t care, for this example)
# Analyzer configuration common to all ports.
# Configure VLAN 1 to contain port 8.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 0
VLANACCESS.VLAN_PORT_MASK = 0x00000100
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 2 to contain port 8.
VLANTIDX.INDEX = 2
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 0
VLANACCESS.VLAN_PORT_MASK = 0x00000100
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 3 to contain ports 9 and 10.
VLANTIDX.INDEX = 3
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 0
VLANACCESS.VLAN_PORT_MASK = 0x00000600
VLANACCESS.VLAN_TBL_CMD = 2
```

## 6.4.6 Spanning Tree Protocols

This section provides information about Rapid Spanning Tree Protocol (RSTP) support and Multiple Spanning Tree Protocol (MSTP) support. The devices also support legacy Spanning Tree Protocol (STP). STP was obsoleted by RSTP in IEEE 802.1D and is not described in this document.

It is assumed that only LAN ports connected to the switch core participate in the spanning tree protocol. This implies that BPDUs are terminated by the switch core.

### 6.4.6.1 Rapid Spanning Tree Protocol

The following table lists the analyzer configuration registers for Rapid Spanning Tree Protocol (RSTP) operation.

***Table 169 •*** **Analyzer Configurations for RSTP Support**

| Register/Register Field | Description | Replication |
|---|---|---|
| PGID[80-106] | Source masks used for ingress filtering | Per port |
| PGID[64-79] | Aggregation masks that can be used for egress filtering for RSTP | 16 |
| PORT_CFG.LEARN_ENA | Enable learning per port | Per port |

*Table 169 •* **Analyzer Configurations for RSTP Support *(continued)***

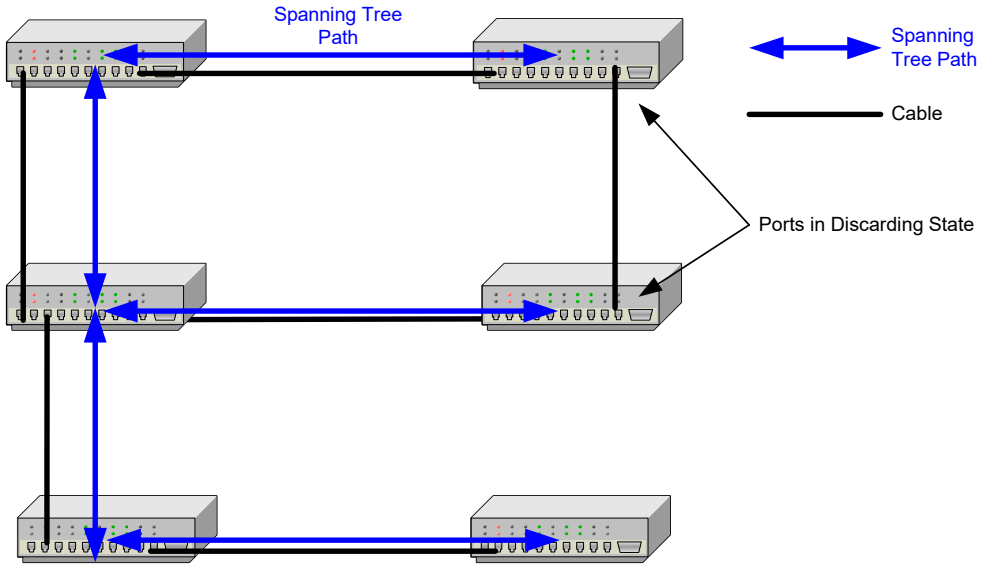
| Register/Register Field | Description | Replication |
|---|---|---|
| CPU_FWD_BPDU_CFG | Enable redirection of frames with reserved BPDU DMAC addresses | Per port per address |
| CPUQ_8021_CFG.CPUQ_BPDU_VAL | CPU extraction queue for redirected BPDU frames | Per address |

To eliminate potential loops in a network, the Rapid Spanning Tree Protocol in IEEE 802.1D creates a single path between any two bridges in a network, adding stability and predictability to the network. The protocol is implemented by assigning states to all ports. Each state controls a port's functionality, limiting its ability to receive and transmit frames and learn addresses.

Establishing a spanning tree is done through the exchange of BPDUs between bridge entities. BPDUs are frequently exchanged between neighboring bridges. These frames are identified by the Bridge protocol address range (DMAC = 01-80-C2-00-00-0x).

When there is a change in the network topology, the protocol reconfigures the port states.

*Figure 73 •* **Spanning Tree Example**




The following table lists the Rapid Spanning Tree port state properties.

*Table 170 •* **RSTP Port State Properties**

| State | BPDU Reception | BPDU Generation | Frame Forwarding | SMAC Learning |
|---|---|---|---|---|
| Discarding | Yes | Yes | No | No |
| Learning | Yes | Yes | No | Yes |
| Forwarding | Yes | Yes | Yes | Yes |

The legacy STP states disabled, blocking, and listening correspond to the discarding state of RSTP.

All frames with a Bridge protocol address must be redirected to the CPU. This is configured in CPU_FWD_BPDU_CFG. BPDUs are forwarded to the CPU irrespective of the port's RSTP state. CPUQ_8021_CFG.CPUQ_BPDU_VAL can be used to configure in which CPU extraction queue the BPDUs are placed. BPDU generation is done through frame injection from the CPU.

Frame forwarding is controlled through ingress filtering and egress filtering. Ingress filtering can be done by using the source masks (PGID[80-106]), and egress filtering can be done by using the aggregation masks (PGID[64-79]). Forwarding can be disabled for ports not in the Forwarding state by clearing their source masks and excluding them from all aggregation masks. The use of the aggregation masks for egress filtering does not preclude the combination of link aggregation and RSTP support. All ports in a link aggregation group that are not in the Forwarding state must be disabled in all aggregation masks. For link aggregated ports in the Forwarding state, the aggregation masks must be configured for link aggregation (such as when RSTP is not supported.)

Learning can be enabled per port with the PORT_CFG.LEARN_ENA.

The following table provides an overview of the port state configurations for port p.

***Table 171 •*** **RSTP Port State Configuration for Port p**

| State | CPU_FWD_BPDU_CFG[p].BPDU_REDIR_ENA[0] | PGID[80+p] | PGID[64-79], All 16 Masks, Bit p | PORT_CFG[p].LEARN_ENA |
|---|---|---|---|---|
| Discarding | 1 | 0 | 0 | 0 |
| Learning | 1 | 0 | 0 | 1 |
| Forwarding | 1 | 1 except for bit p | 1 | 1 |

#### 6.4.6.1.1 RSTP Example

**Situation:**

Port 0 is in the RSTP Discarding state. Port 2 is in the RSTP Learning state. Port 3 is in the RSTP Forwarding state. All other ports on the switch are unused.

**Resolution:**

```
# Get Spanning Tree Protocol BDPUs to CPU extraction queue 0 for port 0, 2,
and 3.
CPU_FWD_BPDU_CFG[0].BPDU_REDIR_ENA[0] = 1
CPU_FWD_BPDU_CFG[2].BPDU_REDIR_ENA[0] = 1
CPU_FWD_BPDU_CFG[3].BPDU_REDIR_ENA[0] = 1
CPUQ_8021_CFG.CPUQ_BPDU_VAL[0] = 0
# Configure the source mask for port 0 (Discarding state).
PGID[80] = 0x00
# Configure the source mask for port 2 (Learning state).
PGID[82] = 0x00
# Configure the source mask for port 3 (Forwarding state).
PGID[83] = 0x77
# Configure the aggregation masks to only allow forwarding to port 3
# (Forwarding state).
PGID[64-79] = 0x08
# Configure the learn mask to only allow learning on ports
# 2 (Learning state) and 3 (Forwarding state).
PORT_CFG[0].LEARN_ENA = 0
PORT_CFG[2].LEARN_ENA = 1
PORT_CFG[3].LEARN_ENA = 1
```

### 6.4.6.2 Multiple Spanning Tree Protocol

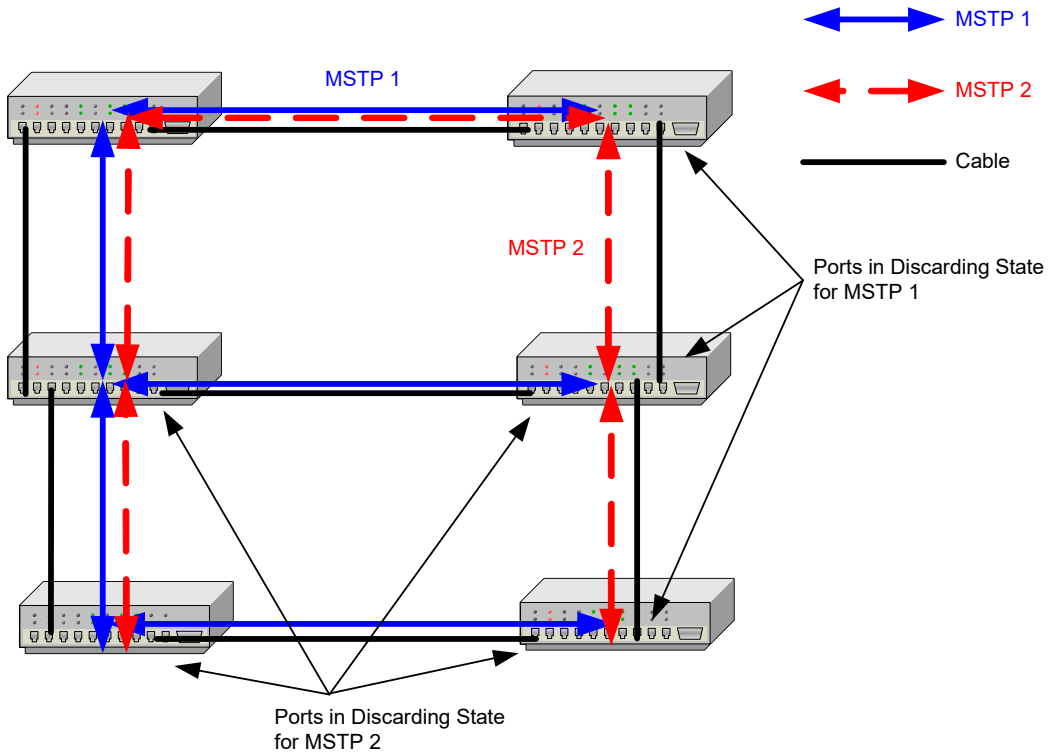
The following table lists the analyzer configuration registers for Multiple Spanning Tree Protocol (MSTP) operation.

***Table 172 •*** **Analyzer Configurations for MSTP Support**

| Register/Register Field | Description | Replication |
|---|---|---|
| VLANACCESS.VLAN_SRC_CHK | Per-VLAN ingress filtering enable. Part of VLAN table command for indirect access to configuration of the 4095 VLANs | None |
| VLANMASK | Per-port VLAN ingress filtering enable | None |
| ADVLEARN.VLAN_CHK | Disable learning for frames discarded due to VLAN membership source port filtering | None |
| PORT_CFG.LEARN_ENA | Enable learning per port | Per port |
| CPU_FWD_BPDU_CFG | Enable redirection of frames with reserved BPDU DMAC addresses | Per port per address |
| CPUQ_8021_CFG.CPUQ_BPDU_VAL | CPU extraction queue for redirected BPDU frames | Per address |

The Multiple Spanning Tree Protocol (MSTP) in IEEE 802.1Q increases network use, relative to RSTP, by creating multiple spanning trees that VLANs can map to independently, rather than having only one path between bridges common for all VLANs. The multiple spanning trees are created by assigning different bridge identifiers for each spanning tree. Mapping the VLANs to spanning trees is done arbitrarily.

***Figure 74 •*** **Multiple Spanning Tree Example**



The Learning state is not supported for MSTP. However, this has limited impact, because when the port is taken to the Forwarding state, learning is done at wire-speed, and, as a result, the SMAC learn delay is less important. MSTP is supported for all VLANs.

The following table lists the multiple spanning tree port state properties.

***Table 173 •*** **MSTP Port State Properties**

| State per VLAN | BPDU Reception | BPDU Generation | Frame Forwarding | SMAC Learning |
|---|---|---|---|---|
| Discarding | Yes | Yes | No | No |
| Learning (not supported) | Yes | Yes | No | Yes |
| Forwarding | Yes | Yes | Yes | Yes |

To enable the MSTP port states:

- Ensure that the switch is VLAN-aware. For more information, see Standard VLAN Operation, page 197.
- Set the ADVLEARN.VLAN_CHK bit to prevent learning of frames discarded due to VLAN ingress filtering.
- Configure all ports as defined for the forwarding state of the RSTP port. For more information, see Table 171, page 212.

Port states per VLAN are hereafter solely configured through the VLAN masks as listed in the following table for port p and VLAN v.

***Table 174 •*** **MSTP Port State Configuration for Port p and VLAN v**

| State | VLAN_ACCESS. VLAN_SRC_CHKVLAN v | VLAN_ACCESS. VLAN_PORT_MASK Bit p, VLAN v |
|---|---|---|
| Discarding | 1 | 0 |
| Learning | Not supported | Not supported |
| Forwarding | 1 | 1 |

As an alternative to setting the VLANACCESS.VLAN_SRC_CHK bit in all VLAN entries in the VLAN table, VLAN ingress filtering can be enabled globally for all VLANs on a per port basis through VLANMASK.

For all multiple spanning tree instances, BPDUs are forwarded to the CPU irrespective of the port states.

#### 6.4.6.2.1 MSTP Example

**Situation:**

Ports 10 and 11 are both members of VLANs 20 and 21. Two spanning trees are used:

- Spanning tree for VLAN 20, where both ports 10 and 11 are in the Forwarding state
- Spanning tree for VLAN 21, where port 10 is in the Discarding state and port 11 is in the Forwarding state

All other ports on the switch are unused.

**Resolution:**

```
# Get all BDPUs to CPU queue 0.
CPU_FWD_BPDU_CFG[*].BPDU_REDIR_ENA[0] = 1
CPUQ_8021_CFG.CPUQ_BPDU_VAL[0] = 0
# Enable learning on all ports. The VLAN table controls forwarding and
learning.
PORT::PORT_CFG.LEARN_ENA = 1
# Disable learning of VLAN membership source port filtered frames.
ADVLEARN.VLAN_CHK = 1
```

```
# Configure VLAN 20 for ports 10 and 11 in Forwarding state.
VLANTIDX.INDEX = 20
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x00000C00
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 21 for port 10 in Discarding state and port 11 in Forwarding
state.
VLANTIDX.INDEX = 21
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x00000800
VLANACCESS.VLAN_TBL_CMD = 2
```

## 6.4.7 IEEE 802.1X: Network Access Control

IEEE 802.1X Port-Based Network Access Control provides a standard for authenticating and authorizing devices attached to a LAN port.

Generally, IEEE 802.1X is port-based; however, the devices also support MAC-based network access control.

This section provides information about the configuration settings for port-based and MAC-based network access control.

### 6.4.7.1 Port-Based Network Access Control

The following table lists the configuration settings that are required for port-based network access control.

***Table 175 •*** **Configurations for Port-Based Network Access Control**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::CPU_FWD_BPDU_CFG.BPDU_REDIR_ENA[3] | Must be set to 1 to redirect frames with destination MAC addresses 01-80-C2-00-00-03 to the CPU Port Module.<br>IEEE 802.1X uses MAC address 01-80-C2-00-00-03. | Per port |
| ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL[3] | Queue to which authentication BPDUs are redirected. | None |
| ANA::PGID[64-79] | When a port is not yet authenticated, any forwarding of frames to the port can be disabled by clearing the port’s bit in all 16 aggregation masks.<br>After authenticated, these bits must be set. | 16 |

*Table 175 •* **Configurations for Port-Based Network Access Control *(continued)***

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::PGID[80-106] | Source masks.<br>When a port is not yet authenticated, any forwarding of frames received on the port must be disabled. This can be done by setting the ANA::PGID[80+port] to all-zeros.<br>After authenticated, the port's source mask must be set back to its normal value. | Per port |

The configuration settings required for port-based network access control enable the following functionality:

- Redirects frames with DMAC 01-80-C2-00-00-03 to CPU, even if the port is not yet authenticated.
- Stops forwarding of frames to ports that are not yet authenticated. This is configured in ANA::PGID[64-79].
- Stops forwarding of frames received on ports that are not yet authenticated. This is configured in ANA::PGID[80-106].

### 6.4.7.2 MAC-Based Authentication with Secure CPU-Based Learning

The following table lists the configuration settings required for MAC-based network access control with secure CPU-based learning.

*Table 176 •* **Configurations for MAC-Based Network Access Control with Secure CPU-Based Learning**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA:PORT:CPU_FWD_BPDU_CFG.BPDU_REDIR_ENA[3] | Must be set to 1 to redirect frames with destination MAC addresses 01-80-C2-00-00-03 to the CPU Port Module.<br>IEEE 802.1X uses MAC address 01-80-C2-00-00-03. | Per port |
| ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL[3] | Queue to which authentication BPDUs are redirected. | None |
| ANA:PORT:PORT_CFG.LEARN_ENA<br>ANA:PORT:PORT_CFG.LEARNCPU<br>ANA:PORT:PORT_CFG.LEARNDROP<br>ANA:PORT:PORT_CFG.LEARNAUTO | Must be set to support secure CPU-based learning. See Address Learning, page 196.<br>PORT_CFG.LEARN_ENA = 1<br>PORT_CFG.LEARNCPU = 1<br>PORT_CFG.LEARNDROP = 1<br>PORT_CFG.LEARNAUTO = 0 | Per port |

The MAC-based network access control with secure CPU-based learning enables the following functionality:

- Redirects frames with DMAC 01-80-C2-00-00-03 to CPU.
- Only frames from known, authenticated MAC addresses are forwarded to other ports.
- Frames from unknown MAC addresses are redirected to CPU for authentication. After the address is authenticated, the CPU must insert an entry in the MAC table. The authentication process may be initiated from the CPU when receiving learn frames.

### 6.4.7.3 MAC-Based Authentication with No Learning

The following table lists the configuration settings required for MAC-based network access control with no learning.

***Table 177 •*** **Configurations for MAC-Based Network Access Control with No Learning**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA:PORT:CPU_FWD_BPDU_CFG .BPDU_REDIR_ENA[3] | Must be set to 1 to redirect frames with destination MAC addresses 01-80-C2-00-00-03 to the CPU Port Module.<br>IEEE 802.1X uses MAC address 01-80-C2-00-00-03. | Per port |
| ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL[3] | Queue to which authentication BPDUs are redirected. | None |
| ANA:PORT:PORT_CFG.LEARN_ENA<br>ANA:PORT:PORT_CFG.LEARNCPU<br>ANA:PORT:PORT_CFG.LEARNDROP<br>ANA:PORT:PORT_CFG.LEARNAUTO | Must be set to support no learning. See Address Learning, page 196.<br>PORT_CFG.LEARN_ENA = 1<br>PORT_CFG.LEARNCPU = 1<br>PORT_CFG.LEARNDROP = 1<br>PORT_CFG.LEARNAUTO = 0 | None |

The MAC-based network access control with no learning enables the following functionality:

- Frames with DMAC 01-80-C2-00-00-03 are redirected to CPU. Unauthenticated and unauthorized devices must initiate an 802.1X session by sending 802.1X BPDUs (MAC address: 01-80-C2-00-00-03). After the address is authenticated, the CPU must insert an entry in the MAC table.
- Only frames from known, authenticated MAC addresses are forwarded to other ports.
- Frames from unknown MAC addresses are discarded and the CPU can therefore not initiate the authentication process.

## 6.4.8 Link Aggregation

Link aggregation bundles multiple ports (member ports) together into a single logical link. It is primarily used to increase available bandwidth without introducing loops in the network and to improve resilience against faults. A link aggregation group (LAG) can be established with individual links being dynamically added or removed. This enables bandwidth to be incrementally scaled based on changing requirements. A link aggregation group can be quickly reconfigured if faults are identified.

Frames destined for a LAG are sent on only one of the LAG's member ports. The member port on which a frame is forwarded is determined by a 4-bit aggregation code (AC) that is calculated for the frame.

The aggregation code ensures that frames belonging to the same frame flow (for example, a TCP connection) are always forwarded on the same LAG member port. For that reason, reordering of frames within a flow is not possible. The aggregation code is based on the following information:

- SMAC
- DMAC
- Source and destination IPv4 address.
- Source and destination TCP/UDP ports for IPv4 packets
- Source and destination TCP/UDP ports for IPv6 packets
- IPv6 Flow Label

For best traffic distribution among the LAG member ports, enable all six contributions to the aggregation code.

Each LAG can consist of up to 16 member ports. Any quantity of LAGs may be configured for the device (only limited by the quantity of ports on the device.) To configure a proper traffic distribution, the ports within a LAG must use the same link speed.

A port cannot be a member of multiple LAGs.

### 6.4.8.1 Link Aggregation Configuration

The following table lists the registers associated with link aggregation groups.

***Table 178 •*** **Link Aggregation Group Configuration Registers**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::PGID[0 – 63] | Destination mask | 64 |
| ANA::PGID[80 – 106] | Source mask. | Per port |
| ANA::PGID[64 – 79] | Aggregation mask. | 16 |
| ANA::PORT_CFG.PORTID_VAL | Logical port number. Must be set to the same value for all ports that are part of a given LAG; for example, the lowest port number that is a member of the LAG. | Per port |
| ANA::AGGR_CFG.AC_IP6_FLOW_LBL_ENA | Use IPv6 flow label when calculating AC. Configure identically for all ports.<br>Recommended value is 1. | None |
| ANA::AGGR_CFG.AC_SIPDIP_ENA | Use IPv4 source and destination IP address when calculating aggregation code. Configure identically for all ports.<br>Recommended value is 1. | None |
| ANA::AGGR_CFG.AC_TCPUDP_PORT_ENA | Use IPv4 TCP/UDP port when calculating aggregation code. Configure identically for all ports.<br>Recommended value is 1. | None |
| ANA:: AGGR_CFG.AC_DMAC_ENA | Use destination MAC address when calculating aggregation code. Configure identically for all ports.<br>Recommended value is 1. | None |
| ANA:: AGGR_CFG.AC_SMAC_ENA | Use source MAC address when calculating aggregation code. Configure identically for all ports.<br>Recommended value is 1. | None |
| ANA:: AGGR_CFG.AC_RND_ENA | Use random aggregation code.<br>Recommended value is 0. | None |

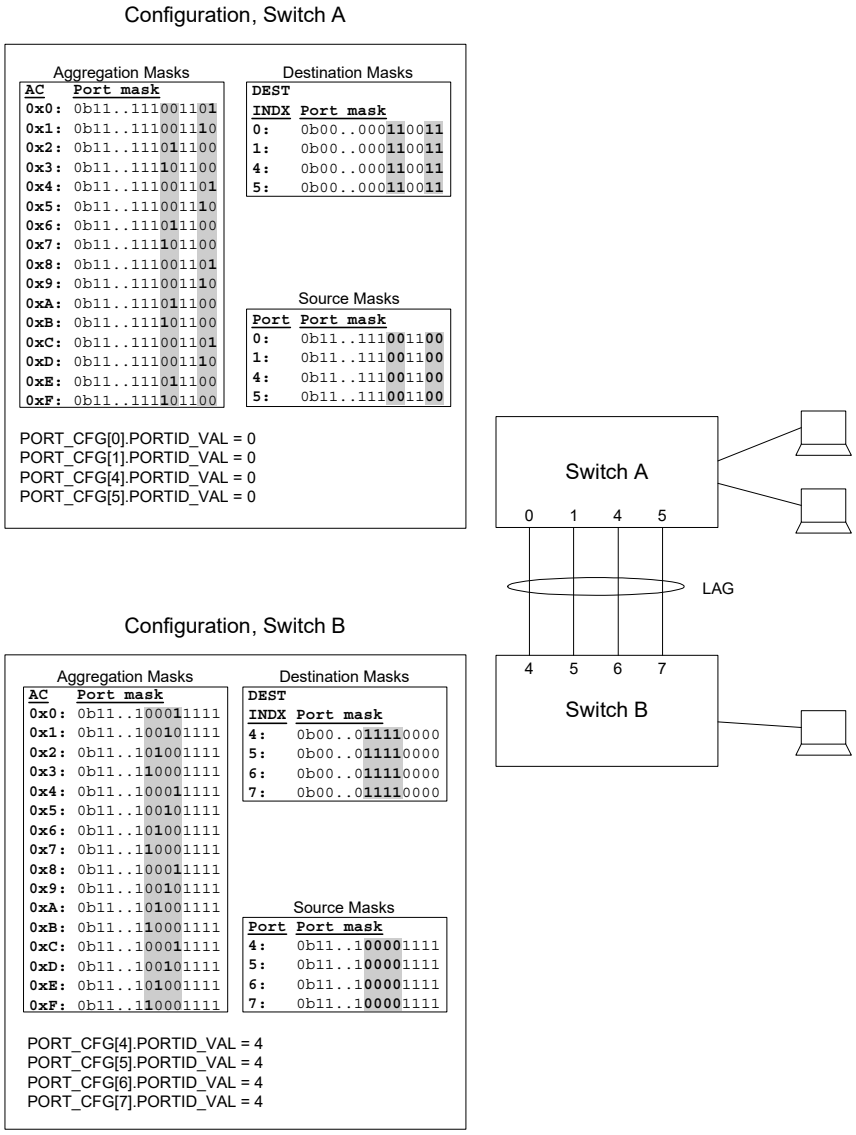
To set up a link aggregation group, the following destination masks, source masks, and aggregation masks must be configured:

- **Destination Masks: ANA::PGID[0-63]** — For each of the member ports, the corresponding destination mask must be configured to include all member ports of the LAG.
- **Source Masks: ANA::PGID[80-106]** — The source masks must be configured to avoid flooding frames that are received at one member port back to another member port of the LAG. As a result, the source masks for each of the member ports must be configured to exclude all of the LAG's member ports.

- **Aggregation Masks: ANA::PGID[64-79]** — The aggregation masks must be configured to ensure that when a frame is destined for the LAG, it gets forwarded to exactly one of the LAG's member ports. Also, the distribution of traffic between member ports is determined by this configuration.

The following illustration shows an example of a LAG configuration.

***Figure 75 •*** **Link Aggregation Example**




In this example, ports 0, 1, 4, and 5 of switch A are configured as a LAG. These ports are connected to 4 ports (4, 5, 6, 7) of switch B, providing an aggregated bandwidth of 4 Gbps between the two switches.

The aggregation masks for switch A are configured such that frames (destined for the LAG) are distributed on the member ports as follows:

- Port 0 if frame's aggregation code (AC) is 0x0, 0x4, 0x8, 0xC
- Port 1 if frame's aggregation code (AC) is 0x1, 0x5, 0x9, 0xD
- Port 4 if frame's aggregation code (AC) is 0x2, 0x6, 0xA, 0xE
- Port 5 if frame's aggregation code (AC) is 0x3, 0x7, 0xB, 0xF

### 6.4.8.2 Link Aggregation Control Protocol (LACP)

LACP allows switches connected to each other to automatically discover if any ports are member of the same LAG.

To implement LACP, any LACP frames must be redirected to the CPU. Such frames are identified by the DMAC being equal to 01-80-C2-00-00-02 (Slow Protocols Multicast address).

The following table lists the registers associated with configuring the redirection of LACP frames to the CPU.

***Table 179 •*** **Configuration Registers for LACP Frame Redirection to the CPU**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::CPU_FWD_BPDU_CFG.BPDU_REDIR_ENA[2] | Must be set to 1. | Per port |

## 6.4.9 Simple Network Management Protocol (SNMP)

This section provides information about the port module registers and the analyzer registers for SNMP operation.

The following table lists the system registers for SNMP operation.

***Table 180 •*** **System Registers for SNMP Support**

| Register | Description | Replication |
|---|---|---|
| CNT | The value of the counter.<br>For more information about how to read counters, see Statistics, page 43. | None |

The following table lists the analyzer registers for SNMP support.

***Table 181 •*** **Analyzer Registers for SNMP Support**

| Register | Description | Replication |
|---|---|---|
| MACACCESS | Command register for indirect MAC table access. Supports GET_NEXT command | None |
| MACHDATA | High part of data word when accessing MAC table. | None |
| MACLDATA | Low part of data word when accessing MAC table. | None |
| MACTINDX | Index for direct-mode access to MAC table. | None |

For SNMP support according to IETF RFC 1157, use the following features:

- RMON counters
- MAC table GET_NEXT function

For more information about the supported RMON counters, see Port Counters, page 191.

For more information about the MAC table GET_NEXT function, see Table 59, page 87.

## 6.4.10 Mirroring

To debug network problems, selected traffic can be copied, or mirrored, to a mirror port where a frame analyzer can be attached to analyze the frame flow.

The traffic to be copied to the mirror port can be selected as follows:

- All frames received on a given port (also known as ingress mirroring)
- All frames transmitted on a given port (also known as egress mirroring
- Frames selected through configured VCAP entries
- All frames classified to specific VIDs
- All frames sent to the CPU (may be useful for software debugging)
- Frames where the source MAC address is to be learned (also known as learn frame), which may be useful for software debugging

The mirror port may be any port on the device, including the CPU.

### 6.4.10.1 Mirroring Configuration

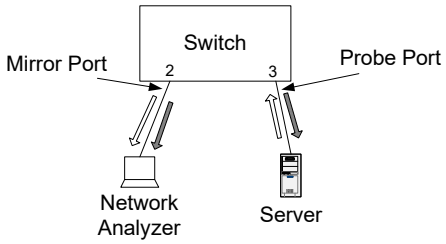
The following table lists configuration registers associated with mirroring.

***Table 182 •*** **Configuration Registers for Mirroring**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::PORT_CFG.SRC_MIRROR_ENA | If set, all frames received on this port are mirrored to the port set configured in MIRRORPORTS, that is, ingress mirroring. | Per port |
| ANA:: EMIRRORPORTS | Frames forwarded to ports in this mask are mirrored to the port set configured in MIRRORPORTS, that is, egress mirroring. | Per port |
| ANA::VLANTIDX.VLAN_MIRROR | If set, all frames classified to this VLAN are mirrored to the port set configured in MIRRORPORTS. | One per VID |
| ANA::AGENCTRL.MIRROR_CPU | Frames destined for the CPU extraction queues are also forwarded to the port set configured in MIRRORPORTS. | None |
| ANA::MIRRORPORTS | The mirror ports. Usually only one mirror port is configured, that is, only one bit is set in this mask. | None |
| ANA::CPUQ_CFG.CPUQ_MIRROR | CPU extraction queue used, if CPU is included in MIRRORPORTS. | None |
| ANA::ADVLEARN.LEARN_MIRROR | Learn frames are also forwarded to ports marked in MIRRORPORTS. | None |
| VCAP Registers | Configuration of VCAP entries, for example, to trigger copy to mirror port. For more information, see VCAP IS2, page 67. | Per VCAP entry |

The following illustration shows a port mirroring example.

***Figure 76 •*** **Port Mirroring Example**



All traffic to and from the server on port 3 (the probe port) is mirrored to port 2 (the mirror port). Note that the mirror port may become congested, because both the Rx frames and Tx frames on the probe port become Tx frames on the mirror port. The following mirror configuration is required:

ANA::PORT_CFG[3].SRC_MIRROR_ENA = 1
ANA::EMIRRORPORTS[3] = 1
ANA::MIRRORPORTS = 0x0000004

In addition to the mirror configuration settings, the egress configuration of the mirror port (port 2) must be configured identically to the egress configuration of the probe port (port 3). This is to ensure that VLAN tagging and DSCP remarking at the mirror port is performed consistently with that of the probe port, such that the frame copies at the mirror port are identical to the original frames on the probe port.

Multiple mirror conditions, such as mirror multiple probe ports, VLANs, and so on, can be enabled concurrently to the same mirror port. However, in such configurations, it may not be possible to configure the egress part of the mirror port to perform tagging and DSCP remarking consistent with that of the original frame.

## 6.5 IGMP and MLD Snooping

This section provides information about the features and configurations related to Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) snooping.

By default, Layer-3 multicast data traffic is flooded in a Layer-2 network in the broadcast domain spanned by the VLAN. This causes unnecessary traffic in the network and extra processing of unsolicited frames in hosts not listening to the multicast traffic. IGMP and MLD snooping enables a Layer-2 switch to listen to IGMP and MLD conversations between host and routers. The switch can then prune multicast traffic from ports that do not have a multicast listener, and as a result, do not need a copy of the multicast frame. This is done by managing the multicast group addresses and the associated port masks.

IGMP is used to manage IPv4 multicast memberships, and MLD is used to manage IPv6 multicast memberships.

The devices support IGMPv2/v3 and MLDv1/v2. IGPMv2 and MLDv1 use any-source multicasting (ASM), where the multicast listener joins a group and can receive the multicast traffic from any source. IGMPv3 and MLDv2 introduce source-specific multicasting (SSM), where both source and group are specified by the multicast listener when joining a group.

The support in the devices is two-fold:

- Control plane: IGMP and MLD frames are redirected to the CPU. This enables the CPU to listen to the queries and reports.
- Data plane: By monitoring the multicast group registrations and de-registrations signaled through the IGMP and MLD frames, the CPU can setup multicast group addresses and associated ports.

### 6.5.1 IGMP and MLD Snooping Configuration

To implement IGMP and MLD snooping, any IGMP or MLD frames must be redirected to the CPU. For information about by the conditions by which such frames are identified, see CPU Forwarding Determination, page 57. IGMP and MLD frames can be independently snooped and assigned individual CPU extraction queues.

The following table lists the registers associated with configuring the redirection of IGMP and MLD frames to the CPU.

***Table 183 •*** **Configuration Registers for IGMP and MLD Frame Redirection to CPU**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::CPU_FWD_CFG.IGMP_REDIR_ENA | Must be set to 1 to redirect IGMP frames to the CPU | Per port |
| ANA::CPU_FWD_CFG.MLD_REDIR_ENA | Must be set to 1 to redirect MLD frames to the CPU | Per port |

*Table 183 •* **Configuration Registers for IGMP and MLD Frame Redirection to CPU *(continued)***

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA::CPUQ_CFG.CPUQ_IGMP | CPU extraction queue for IGMP frames | None |
| ANA::CPUQ_CFG.CPUQ_MLD | CPU extraction queue for MLD frames | None |

## 6.5.2 IP Multicast Forwarding Configuration

The following table lists the registers associated with configuring the multicast group addresses and the associated ports.

*Table 184 •* **IP Multicast Configuration Registers**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| MACHDATA | MAC address and VID when accessing the MAC table. | None |
| MACLDATA | MAC address when accessing the MAC table. | None |
| MACTINDX | Direct address into the MAC table for direct read and write. | None |
| MACACCESS | Flags and command when accessing the MAC table. | None |
| MACTOPTIONS | Flags when accessing the MAC table | None |
| FLOODING_IPMC | Index into the PGID table used for flooding of IPv4/6 multicast control and data frames. | None |
| PGID[63:0] | Destination and flooding masks table | 64 |
| IS1_ACTION.FID_SEL | Specifies the use of IS1_ACTION.FID_VAL for the DMAC lookup, the SMAC lookup, or for both lookups. | Per IS1 entry |
| IS1_ACTION.FID_VAL | FID value. | Per IS1 entry |

IPv4 and IPv6 multicast group addresses are programmed in the MAC table as IPv4 and IPv6 multicast entries. For more information, see MAC Table, page 85. The entry in the MAC table also holds the set of egress ports associated with the group address.

By default, programming an IPv4 or IPv6 multicast entry in the MAC table makes it an any-source multicast, because the actual source IP address is insignificant with respect to forwarding.

To create source-specific IPv4 or IPv6 multicast entries, the Filter Identifier (FID) action in VCAP IS1 can be used, which enables creation of specific FIDs per source IP address. Multiple MAC table entries holding the same IPv4 or IPv6 multicast group address but different FIDs can then be created. This effectively enables source-specific multicasting.

The switch provides full control of flooding of unknown IP multicast frames. For more information, see Table 69, page 94. Generally, an IGMP and MLD snooping switch disables flooding of unknown multicast frames, except to ports connecting to multicast routers. Note that unknown IPv4 multicast control frames should be flooded to all ports, because IPv4 is not as strict as IPv6 in terms of registration for IP multicast groups.

# 6.6 Quality of Service (QoS)

This section discusses features and configurations related to QoS.

The devices include a number of features related to providing low-latency guaranteed services to critical network traffic such as voice and video in contrast to best-effort traffic such as web traffic and file transfers.

All incoming frames are classified to a QoS class, which is used in the queue system when assigning resources, in the arbitration from ingress to egress queues and in the egress scheduler when selecting the next frame for transmission. The devices provide two methods for classifying to a QoS class and for remarking priority information in the frame: Basic and Advanced classification.

Basic QoS classification enables predefined schemes for handling Priority Code Points (PCP), Drop Eligible Indicator (DEI), and Differentiated Service Code Points (DSCP):

- QoS classification based on PCP and DEI for tagged frames. The mapping table from PCP and DEI to QoS class is programmable per port.
- QoS classification based on DSCP values. Can optionally use only trusted DSCP values. The mapping table from DSCP value to QoS class is common between all ports.
- The devices have the option to work as a DS boundary node connecting two DS domains together by translating incoming/outgoing DSCP values for selected ports.
- The DSCP values can optionally be remarked based on the frame's classified QoS class.
- For untagged or non-IP frames, a default per-port QoS class is programmable.

Advanced QoS classification uses the VCAP IS1, which provides a flexible classification:

- A large range of higher layer protocol fields (Layer 2 through Layer 4) are available for rule matching.
- The IS1 action vector returns a QoS class, and translations of PCP, DEI, and DSCP values are also possible.
- Through programming of entries in IS1, QoS rules can be made as specific as needed.For example; per source MAC address, per TCP/UDP destination port number, or combination of both.

For more information about advanced QoS classification using the VCAP IS1, see Ingress Control Lists, page 229.

## 6.6.1 Basic QoS Configuration

The following table lists the registers associated with configuring basic QoS.

***Table 185 •*** **Basic QoS Configuration Registers**

| Register | Description | Replication |
|---|---|---|
| ANA:PORT:QOS_CFG | QoS and DSCP configuration | Per port |
| ANA:PORT:QOS_PCP_DEI_MAP_CFG: | Mapping of DEI and PCP to QoS class and drop precedence level | Per port |
| ANA::DSCP_CFG | DSCP configuration | Per DSCP |

**Situation:**

Assume a configuration with the following requirements:

- All frames with DSCP=7 must get QoS class 7.
- All frames with DSCP=8 must get QoS class 5.
- DSCP=9 is untrusted and all frames with DSCP=9 should be treated as a non-IP frame.
- VLAN-tagged frames with PCP=7 must get QoS class 7
- All other IP frames must get QoS class 1.
- All other non-IP frames must get QoS class 0.

**Solution:**

```
# Program overall QoS configuration
QOS_CFG.QOS_DSCP_ENA = 1
QOS_CFG.QOS_PCP_ENA = 1
```

```
# Program DSCP trust configuration (“*” = 0 through 63)
DSCP_CFG[*].DSCP_TRUST_ENA = 1
DSCP_CFG[9].DSCP_TRUST_ENA = 0

# Program DSCP QoS configuration (“*” = 0 through 63)
DSCP_CFG[*].QOS_DSCP_VAL = 1
DSCP_CFG[7].QOS_DSCP_VAL = 7
DSCP_CFG[8].QOS_DSCP_VAL = 5

# Program PCP QoS configuration (“*” = 0 through 15)
# Note: both 7 and 15 are programmed in order to don’t care DEI
QOS_PCP_DEI_MAP_CFG[*] = 0
QOS_PCP_DEI_MAP_CFG[7] = 7
QOS_PCP_DEI_MAP_CFG[15] = 7

# Program default QoS class for non-IP, non-tagged frames.
QOS_CFG.QOS_DEFAULT_VAL = 0
```

### 6.6.2 IPv4 and IPv6 DSCP Remarking

IPv4 and IPv6 packets include a 6-bit Differentiated Services Code Point (DSCP), which switches and routers can use to determine the QoS class of a frame. With a proper value in the DSCP field, packets can be prioritized consistently throughout the network. Compared to QoS classification based on user priority, classification based on DSCP provides two main advantages

- DSCP field is already present in all packets (assuming all traffic is IPv4/IPv6).
- DSCP value is preserved during routing and is therefore better suited for end-to-end QoS signaling.

Some hosts may be able to send packets with an appropriate value in the DSCP field, whereas other hosts may not provide an appropriate value in the DSCP field.

For packets without an appropriate value in the DSCP field, the devices can be configured to write a new DSCP value into the frame, based on the QoS class of the frame. For example, the devices may have determined the QoS class based on the VLAN tag priority information (PCP and DEI). After the packet is transmitted by the egress port, the DSCP field can be rewritten with a value based on the QoS class of the frame. Any subsequent routers or switches can then be easily prioritize the frame, based on the rewritten DSCP value.

The DSCP rewriting functionality available in the devices provide flexible, per-ingress port and per-DSCP-value configuration of whether frames should be subject to DSCP rewrite. If it is determined at the ingress port that the DSCP value should be rewritten and to which value, this is then signaled to the egress ports, where the actual change of the DSCP field is done.

Additionally, the IS1 can be programmed to return a DSCP value as part of the action vector. This value overrules the potential DSCP value coming out of the DSCP rewrite functionality described previously. A DSCP value from either the basic classification or the advanced IS1 classification obey the same egress rules for the actual DSCP remarking.

### 6.6.2.1 DSCP Remarking Configuration

The following table lists the configuration registers associated with DSCP remarking.

***Table 186 •*** **Configuration Registers for DSCP Remarking**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| ANA:PORT:DSCP_REWR_CFG | Two-bit DSCP rewrite mode per ingress port:<br>0x0: No DSCP rewrite.<br>0x1: Rewrite only if the frame's current DSCP value is zero.<br>0x2: Rewrite only if the frame's current DSCP value is enabled for remarking in ANA::DSCP_CFG.DSCP_REWR_ENA.<br>0x3: Rewrite DSCP of all frames, regardless of current DSCP value. | Per ingress port |
| ANA::DSCP_CFG.DSCP_REWR_ENA | Enables specific DSCP values for rewrite for ports with DSCP rewrite mode set to 0x2. | Per DSCP |
| ANA::DSCP_REWR_CFG.DSCP_QOS_REWR_VAL | Maps the frame's DP level and QoS class to a DSCP value. | Per DP level and per QoS class |
| REW::DSCP_CFG.DSCP_REWR_CFG | Enables DSCP rewrite for egress port. | Per egress port |
| REW::DSCP_REMAP_CFG | Remap table of DSCP values. | None |

The configuration related to the ingress port controls whether a frame is to be remarked. For each ingress port, a DSCP rewrite mode is configured in ANA:PORT:DSCP_REWR_CFG. This register defines the four different modes as follows:

- 0x0: No DSCP rewrite, that is, never change the received DSCP value.
- 0x1: Rewrite if DSCP is zero. This may be useful if a DSCP value of zero indicates that the host has not written any value to the DSCP field.
- 0x2: Rewrite selected DSCP values. In ANA::DSCP_CFG.DSCP_REWR_ENA specific DSCP values can be selected for rewrite, for example, if only certain DSCP values are allowed in the network.
- 0x3: Rewrite all DSCP values.

After a frame is selected for DSCP rewrite, based on the configuration for the ingress port, the new DSCP value is determined by mapping the QoS class and DP level to a new DSCP value (ANA::DSCP_REWR_CFG.DSCP_QOS_REWR_VAL).

This DSCP value is overruled by IS1 if a hit in IS1 returns an action vector with DSCP_ENA set.

The resulting DSCP value is forwarded to the Rewriter at the egress port, which determines whether to actually write the new DSCP value into the frame (REW::DSCP_CFG.DSCP_REWR_CFG). Optionally, the DSCP value may be translated before written into the frame (REW::DSCP_REMAP_CFG) for applications where the switch acts as an DS boundary node.

When an IPv4 DSCP is rewritten, the IP header checksum is updated accordingly.

## 6.6.3 Voice over IP (VoIP)

This section provides information about QoS in applications with Voice over IP (VoIP).

In a typical workgroup switch application with VoIP phones, both workstations and VoIP phones are connected to the switch. A workstation can be connected through a VoIP phone. Traffic from the workstation is usually untagged, whereas traffic from the VoIP phone may or may not be tagged. The QoS classification mechanism applied on the access port depends on the capabilities of the VoIP phone; these capabilities vary from phone to phone. With different VoIP phone models in the network, different access ports require different QoS classification mechanisms. The access switch can perform QoS classification, depending on the VoIP phone model, to achieve consistent VoIP QoS across the network.

Voice traffic can be identified in different ways:

- **Source MAC address (OUI): Most vendors use a dedicated OUI for VoIP phones.**
- **EtherType**: Legacy phones may use a special EtherType for VoIP.
- **VID**: A special VID used for voice traffic.
- **UDP Port Range**: Voice traffic often uses a well-known port range for the Real-time Transport Protocol (RTP).
- **DSCP or ToS Precedence**: Many phones can set the DSCP value or the ToS precedence bits.
- **Priority Code Point**: Many phones send VLAN tagged frames and can set the priority code point.

All of these identification methods are supported by QoS classification through IS1. They can be used to determine the VoIP traffic's QoS class when entering the switch. For more information about the IS1, see VCAP IS1, page 62.

To ensure consistent QoS across the network, frames can be remarked on the uplink port. Priority Code Points and DSCP values can be remarked based on the QoS class determined by the QCLs. For more information about Priority Code Point and DSCP remarking, see VLAN Editing, page 116, and IPv4 and IPv6 DSCP Remarking, page 225.

Traffic received on the uplink port can usually rely on simple DSCP or PCP QoS classification.

## 6.7 VCAP Applications

This section provides information about Vitesse Content Aware Processor (VCAP) applications for QoS classification, source IP guarding, and access control.

The following table shows the different control lists that the VCAP can be used to build.

***Table 187 •*** **Control Lists and Application**

| Control List | Description |
|---|---|
| Ingress control lists (ICLs) | QoS classification<br>VLAN classification and translation<br>policy association group classification |
| IPv4 source guarding control lists (S4CLs) | IPv4 source guarding |
| IPv6 source guarding control lists (S6CLs) | IPv6 source guarding |
| Access control lists (ACLs) | Access control |
| Egress control lists (ECLs) | Tagging and egress translations |

### 6.7.1 Notation for Control Lists Entries

Setting up a control list typically requires a large amount of register configurations. To maintain the overview of the VCAP functionality, the following control list notations are used. The register configurations are not listed. For more information about the VCAP configurations, see VCAP-II, page 58.

The notation used is:

```
entry_number vcap entry_type {entry_field=value}
→ {action_field=value}
```

Each control entry in the notation consists of:

- The entry number specifying the TCAM address for the specific TCAM
- The VCAP used (IS1, IS2, ES0)
- The entry type (for instance IS1 or MAC_ETYPE).
- Zero, one, or more entry fields with specified values. If no value is supplied, it is assumed that the value is 1.
- The action (indicated with →)
- Zero, one, or more action fields with specified values. If no value is supplied, it is assumed that the value is 1.

All entry fields not listed in the entry part of the control entry are set to don't care.

All action fields not listed in the action part of the control entry are set to zero.

Default actions are special, because they do not have an entry type and a pattern to match:

```
default vcap (first|second) port=value
→ {action_field=value}
```

The notation is illustrated by the following examples.

**Example 1:**

An example of an ACL entry:

```
255 is2 ipv4_other first igr_port_mask=(1<<11) sip=10.10.12.134
→
```

This ACL entry is located in entry number 255. It is matched for the first lookup, and it is part of the port ACL for port 11. The type is ipv4_other, and the action is not to change the normal flow for frames with SIP = 10.10.12.134.

**Example 2:**

Policy ACL A can include a monitoring rule that disables forwarding and learning of all incoming IPv4 traffic, but redirects a copy to CPU extraction queue number 3 using the hit-me-once filter. The hit-me-once filter enables the CPU to control when it ready to accept a new frame. The rule would look like this:

```
254 is2 ipv4_other first pag=A
→ hit_me_once cpu_qu_num = 3
```

**Example 3:**

This example shows an ACE that allows forwarding and learning of ARP requests from port 11, if the source IP address is 10.10.12.134. The ACL entry also performs ARP sanity checks that frames must pass to match. The checks include checking that it is a Layer-2 broadcast, that the hardware address space is Ethernet, that the protocol address space is IP, that the MAC address and IP address lengths are correct, and that the sender hardware address (SMAC) matches the SMAC of the frame.

```
253 is2 arp first igr_port_mask=(1<<11) l2_bc opcode=arp_request
sip=10.10.12.134
arp_addr_space_ok arp_proto_space_ok arp_len_ok
arp_sender_match
→
```

**Example 4:**

If the default action from first lookup for port 11 is to discard all traffic, the following notation is used:

```
default is2 first port=11
→ mask_mode=1 port_mask=0x0
```

## 6.7.2 Ingress Control Lists

The following table lists the registers associated with advanced QoS configuration through Ingress Control Lists.

***Table 188 •*** **Advanced QoS Configuration Register Overview**

| Register | Description | Replication |
|---|---|---|
| ANA:PORT:QOS_CFG | QoS configuration | Per port |

**Situation:**

Assume a configuration with the following requirements:

- All frames with DSCP = 7 must get QoS class 2.
- All frames with TCP/UDP port numbers in the range 0 – 1023 must get QoS class 3, except frames with TCP/UDP port 25, which must get QoS class 1.
- All other frames must get QoS class 0.

**Solution:**

The resulting QoS Control List looks like this:

```
255 is1 is1 first etype_len ip_snap dscp = 7
→ qos_ena=1, qos_val = 2
254 is1 is1 first etype_len ip_snap l4_sport = 25
→ qos_ena=1, qos_val = 1
253 is1 is1 first etype_len ip_snap etype = 25
→ qos_ena=1, qos_val = 1
252 is1 is1 first etype_len ip_snap l4_sport = (key: 0, mask: 0x3FF)
→ qos_ena=1, qos_val = 3
251 is1 is1 first etype_len ip_snap etype = (key: 0, mask: 0x3FF)
→ qos_ena=1, qos_val = 3
```

ANA:PORT:QOS_CFG.QOS_DEFAULT_VAL = 0.

## 6.7.3 Access Control Lists

The examples operate with three levels of ACLs:

- Port ACLs
- Policy ACLs
- Switch ACLs

The port ACLs are specific to a single port or a group of ports that form a link aggregation group. For example, a port ACL can be used for source IP filtering, locking a specific source IP address to a port. For more information about this example, see Restrictive SIP Filter Using IS2, page 231.

The policy ACLs are shared for a group of ports that must have the same policy applied. For example, there could be one policy for ports through which workstations access the network and another policy for ports to which servers are connected.

The switch ACLs apply to all ports of the switch. They specify some general rules that apply to all traffic passing through the switch. The rules can still be rather specific, for example, covering a specific VLAN or a specific IP address.

In the examples, the resulting ACL can include one port ACL, one policy ACL, and the switch ACL. This is determined by the way the ingress port mask (IGR_PORT_MASK) and the policy association group (PAG) are used. For information about IGR_PORT_MASK and PAG, see VCAP IS2, page 67. There are

several ways to use the 8-bit PAG, but in this section, all eight bits are used to point out a policy ACL. The IGR_PORT_MASK points out the port ACL. This permits one port ACL per port and a total of 256 policy ACLs. Note that ports may share the same port ACL and a port by don't caring bits in the port ACL's IGR_PORT_MASK.

Each port has a default PAG assigned to it. The IS1 VCAP can be used to change the value of the PAG based on specific protocol fields matched in the IS1 lookup. The resulting PAG is used in the IS2 VCAP lookup and is matched against the PAG field of the ACL entries.

For an ACL entry in the IS2 VCAP, the PAG and IGR_PORT_MASK use this notation:

PAG = PolicyACL_ID
IGR_PORT_MASK = 1<<PortACL_ID

**Notes** The "<<" operator is the bitwise left shift operator. It shifts the left operand bit-wise to the left the number of positions specified by the right operand.

The IGR_PORT_MASK is a mask so the port number is left-shifted to create the mask.

For an ACL entry that is part of a port ACL for port 8, the PAG would be (*) and IGR_PORT_MASK would be (1<<8) = 0x100. The asterisk is a wildcard, which means that the PolicyACL_ID is a don't-care. For an ACL entry that is part of policy ACL A, the PAG would be (A) and the IGR_PORT_MASK would be (*). In this case, the PortACL_ID is a don't-care.

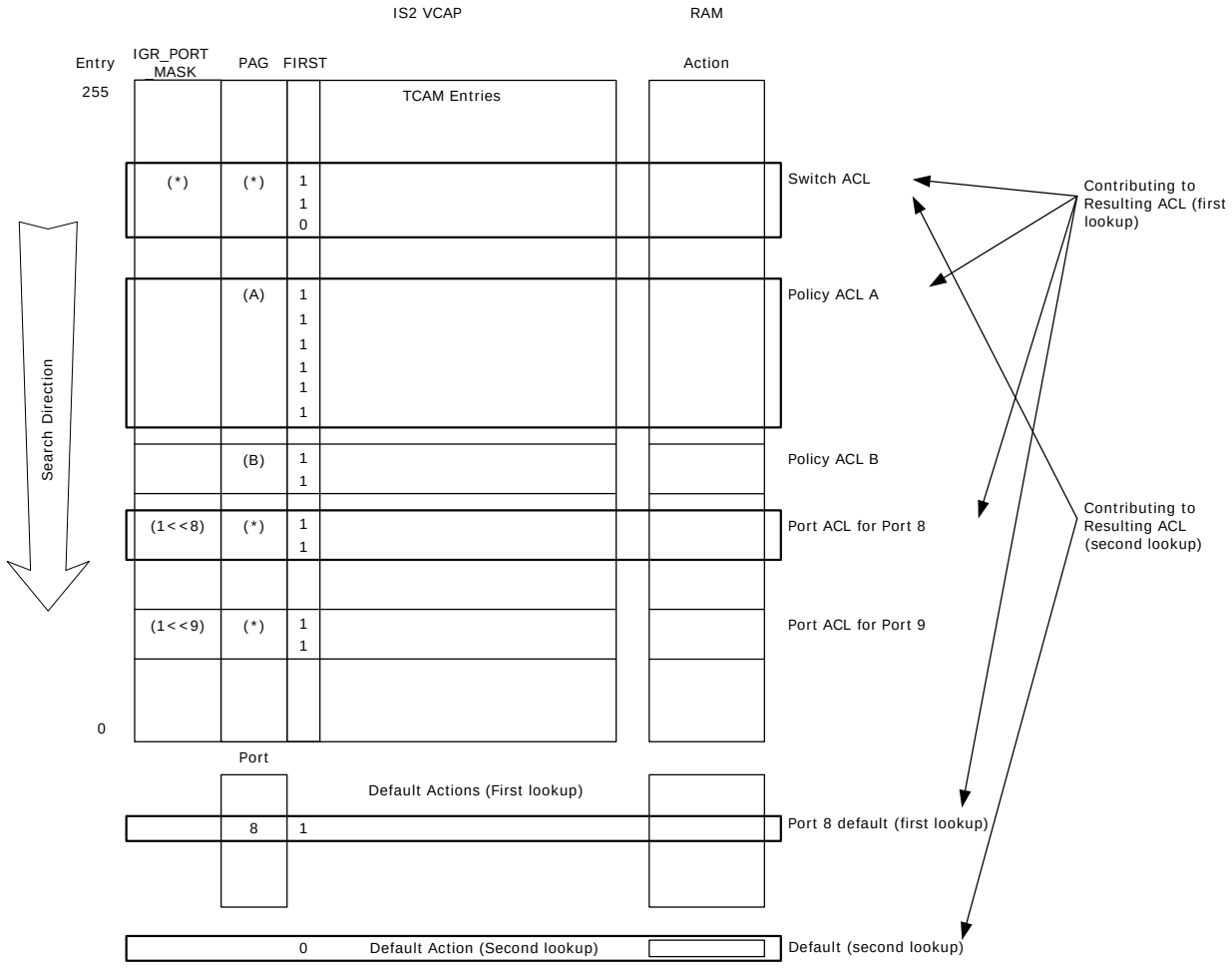
If, for example, port 8 must have policy A applied, the PAG assigned to port 8 is (A). Using this PAG value, the following ACLs match the lookup:

1. The port ACL for port 8 with PAG = (*) and IGR_PORT_MASK = (1<<8)
2. The policy ACL A with PAG = (A) and IGR_PORT_MA
3. SK = (*)
4. The switch ACL with PAG = (*) and IGR_PORT_MASK = (*)

The ordering of the port ACL, the policy ACL, and switch ACL in the resulting ACL follows the ordering in the TCAM. In the following illustration, the switch ACL has the highest priority, followed by the policy ACL A, and finally, the port ACL for port 8.

The resulting ingress ACL in the example is made up of the ingress ACL entries in the switch ACL, the policy ACL A, the port ACL for port 8, and the default action for port 8. The VCAP also does a second lookup, for which the resulting ACL has a common default action as the last rule.

*Figure 77 •* **Resulting ACL for Lookup with PAG = (A) and IGR_PORT_MASK = (1<<8)**




## 6.7.4 Source IP Filter (SIP Filter)

The VCAP enables filtering of source IP (SIP) addresses on a port also known as source IP guarding. This can be used to only allow IP traffic from a specific SIP to enter the switch on a given port. Doing this can prevent the following denial of service (DoS) attacks: LAND attack, SMURF attack, SYN flood attack, Martian attack, and Ping attack.

**Restrictive SIP Filter Using IS2**

A restrictive SIP filter can be applied per port in networks where only IP traffic is allowed. The filter locks a specific SIP to the port and only permits ARP frames and IPv4 frames with the specified SIP to enter the switch on the given port.

For monitoring purposes, it is possible to permit IPv4 frames with other SIPs than the SIP locked to the port. The action is to redirect to the CPU, and the amount of traffic can be reduced by using the hit-me-once feature. The ACL entry for this can be part of a policy ACL for all ports on which the SIP filter is applied.

The port ACL has the following options:

- Permit IPv4 with trusted SIP
- Permit ARP with trusted SIP passing ARP sanity checks
- Permit all IPv4 — CPU redirect with hit-me-once filter (for monitoring)
- Default port action — discard all traffic

**Situation:**

Apply the restrictive SIP filter on port 11 with SIP 10.10.12.134.

**Resolution:**

The resulting ACL for port 11 looks like this:

```
255 is2 ipv4_tcp_udp first igr_port_mask=(1<<11) sip=10.10.12.134
→
254 is2 ipv4_other first igr_port_mask=(1<<11) sip=10.10.12.134
→
253 is2 arp first igr_port_mask=(1<<11) l2_bc opcode=arp_request
sip=10.10.12.134
arp_addr_space_ok arp_proto_space_ok arp_len_ok
arp_sender_match
→
252 is2 ipv4_tcp_upd first pag=A
→ hit_me_once cpu_queue=3
251 is2 ipv4_other first pag=A
→ hit_me_once cpu_queue=3
default is2 first port=11
→ mask_mode=1 port_mask=0x0
```

Applying this SIP filter requires to two entries per port plus three common entries.

**Restrictive SIP Filter Using IS1 and IS2**

The same filter as listed above can be achieved using the host_match actions from IS1.

**Situation:**

Apply the restrictive SIP filter on port 11 with SIP 10.10.12.134.

**Resolution:**

The resulting ACL for port 11 looks like this:

IS1:

```
255 is1 smac_sip4 igr_port=11 sip=10.10.12.134
→ host_match
```

IS2:

```
255 is2 ip4_tcp_upd first host_match=1
→
254 is2 ip4_other first host_match=1
→
253 is2 arp first igr_port_mask=(1<<11) l2_bc opcode=arp_request
sip=10.10.12.134
arp_addr_space_ok arp_proto_space_ok arp_len_ok
arp_sender_match
→
252 is2 ipv4_tcp_udp first pag=A
→ hit_me_once cpu_queue=3
251 is2 ipv4_other first pag=A
→ hit_me_once cpu_queue=3
default is2 first port=11
→ mask_mode=1 port_mask=0x0
```

Applying this SIP filter requires one entry in IS1 per port and five common entries in IS2. This filter can be extended to create a restrictive MAC/IP-binding filter by including the source MAC address in the key in the IS1 smac_sip4 rule.

**Less Restrictive SIP Filter Using IS2**

For networks in which non-IP protocols are allowed, for example IPX and ARP, a less restrictive SIP filter can be applied with the following port ACL:

- Permit IPv4 with trusted SIP
- Discard all IPv4
- Default port action; Permit all traffic (non-IPv4, because all IPv4 traffic is covered by the ACL entries from other two items)

For monitoring purposes, the “Discard all IPv4” ACL can be changed to perform CPU redirect. This allows the CPU to monitor all incoming IPv4 frames with source IP addresses different from the trusted SIP, but without allowing these frames to be forwarded to other ports.

**Situation:**

Apply the less restrictive SIP filter on port 10 with source IP address 10.10.12.134, and monitor any IPv4 traffic with unauthorized source IP addresses with hit-me-once filtering to CPU extraction queue number 2. The monitoring rule is part of policy ACL A that is applied to all user ports.

**Resolution:**

The resulting ingress ACL for port 10 looks like this:

```
255 is2 ipv4_tcp_udp first igr_port_mask=(1<<10) sip=10.10.12.134
→
254 is2 ipv4_other first igr_port_mask=(1<<10) sip=10.10.12.134
→
63 is2 ipv4_tcp_udp first pag=A
→ hit_me_once cpu_queue=2
62 is2 ipv4_other first pag=A
→ hit_me_once cpu_queue=2
default is2 first port=10
→
```

Applying this SIP filter requires two entries per port plus two common entries.

## 6.7.5 DHCP Application

A DHCP application can be supported using one policy ACL for the user ports and another policy ACL for the DHCP server ports.

On the user ports, the DHCP requests must be snooped to be able to automatically reset the SIP filters that are applied per port. DHCP replies should be prevented from being forwarded from user ports. For monitoring purposes, such illegal replies are redirected to the CPU.

On the DHCP server ports, DHCP replies are snooped to be able to automatically update the SIP filter for the user port where the reply goes.

In addition, an egress rule is needed to prevent forwarding of all DHCP requests to user ports.

**Situation:**

Policy ACL A is used for the user port DHCP policy, and policy ACL B is used for the DHCP server policy. The server ports are ports 8 and 9.

Snoop DHCP requests from user ports in CPU extraction queue 1, using policer 0 to protect the CPU. DHCP replies from the servers are snooped in queue 2, and are also subject to policing with policer 0. The illegal DHCP replies from user ports are redirected to queue 3 using the hit-me-once filter.

**Resolution:**

The PAG assigned to the user ports is (A). The PAG assigned to the DHCP server ports (8 and 9) is (B).

The following shows the ACL entries for the DHCP application:

```
255 is2 ipv4_tcp_udp protocol=udp
sport=bootp_client dport=bootp_server
→ mask_mode=1 port_mask=0x0000300
63 is2 ipv4_tcp_udp first pag=A protocol=udp
sport=bootp_client dport=bootp_server
→ cpu_copy_ena cpu_queue=1 police_ena police_idx=0
62 is2 ipv4_tcp_udp first pag=A protocol=udp
```

```
sport=bootp_server dport=bootp_client
→ hit_me_once cpu_queue=3
31 is2 ipv4_tcp_udp first pag=B protocol=udp
sport=bootp_server dport=bootp_client
→ cpu_copy_ena cpu_queue=2 police_ena police_idx=0
default is2 first
→ mask_mode=1 port_mask=0x0
default is2 second
→
```

Regardless of the number of ports covered, four ACL entries are used: one in the switch ACL, two in policy ACL A, and one in policy ACL B.

## 6.7.6 ARP Filtering

The VCAP support two useful ARP filters:

- Policing ARP requests to the switch's IP address to mitigate DoS attacks by ARP flooding
- Performing general ARP sanity checks

Because these are general rules, it is sensible to make them part of the switch ACL.

**Situation:**

Discard all ARP frames that do not pass the ARP sanity checks. Police ARP requests to the switch's IP address 10.10.12.1 using ACL policer 2. ACL policer 2 is configured to allow 16 frames per second, and the frames are copied to CPU extraction queue 0.

RARP is not allowed in the network.

**Resolution:**

To do ARP filtering in the switch ACL, perform the filtering for the switch's IP address first, then allow all ARP frames passing the sanity checks, and finally, discard all remaining ARP frames. This is illustrated by the following:

```
255 is2 arp first l2_bc opcode=arp_request
dip=10.10.12.1
arp_addr_space_ok arp_proto_space_ok arp_len_ok
arp_sender_match
→ cpu_copy_ena cpu_queue=0 police_ena police_idx=255
254 is2 arp first l2_bc opcode=(arp_request or arp_reply)
arp_addr_space_ok arp_proto_space_ok arp_len_ok
arp_sender_match
→
253 is2 arp
→mask_mode=1 port_mask=0x0
```

The ACL policer configuration for policer 255 is done as follows:

```
# Set the base unit to 1 frame per second, enable the policer, and set the rate
to 16 frames per second and a burst of 1 frame:
SYS:POL[255]:POL_MODE_CFG.FRM_MODE = 1
SYS:POL[255]:POL_PIR_CFG.PIR_RATE = 16
SYS:POL[255]:POL_PIR_CFG.PIR_BURST = 3
```

Three ACL entries are used, irrespective of the number of ports covered.

## 6.7.7 Ping Policing

The network can easily be protected against ping attacks using a switch ACL rule that applies an ACL policer to all ping packets.

**Situation:**

Allow no more than 128 ping packets per second to be forwarded through the switch by means of ACL policer 15. Ping packets in excess of 128 frames per second are discarded.

**Resolution:**

Ping packets are ICMP frames with ICMP Type = Echo Request. Echo Request is specified by the first byte of the ICMP frame being 0x08. The rest of the ICMP frame is don't-care. ICMP frames are carried in IPv4 frames with the protocol value 0x01.

The resulting switch ACL entry is as follows:

```
127 is2 ipv4_other first protocol=icmp ip4_payload_high=0x8*
→ police_ena police_idx=15
```

ACL policer 15 in the policer pool is configured to 128 frames per second like this:

SYS:POL[15]:POL_MODE_CFG.FRM_MODE = 1
SYS:POL[15]:POL_PIR_CFG.PIR_RATE = 128
SYS:POL[15]:POL_PIR_CFG.PIR_BURST = 1

One ACE is used, regardless of the number of ports covered.

### 6.7.8 TCP SYN Policing

A server in the network can be protected against TCP SYN DoS attacks by policing TCP connection requests to the server's IP address.

**Situation:**

Allow no more than 128 new TCP connections per second to the server with IP address 10.10.12.99. Use ACL policer 5.

**Resolution:**

TCP connection requests are TCP frames with the SYN flag set. The resulting switch ACL entry is as follows:

```
127 is2 ipv4_tcp_udp first protocol=tcp
dip=10.10.12.99
syn
→ police_ena police_idx=5
```

ACL policer 5 in the policer pool is configured to 128 frames per second by the following:

SYS:POL[5]:POL_MODE_CFG.FRM_MODE = 1
SYS:POL[5]:POL_PIR_CFG.PIR_RATE = 128
SYS:POL[5]:POL_PIR_CFG.PIR_BURST = 1

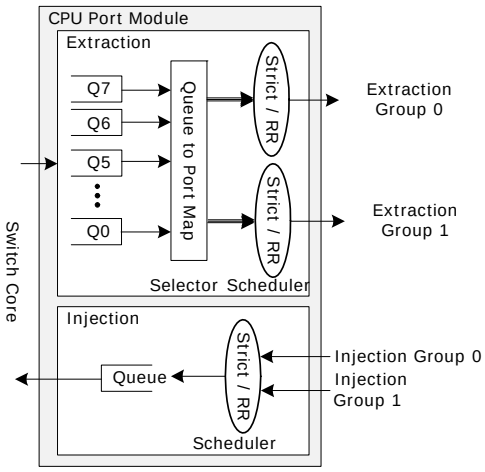
One ACE is used, regardless of the number of ports covered.

## 6.8 CPU Extraction and Injection

This section provides information about how the CPU extracts and injects frames to and from the switch core.

The following illustration shows the CPU Port Module used for injection and extraction.

*Figure 78 •* **CPU Extraction and Injection**




The switch core forwards CPU extracted frames to eight CPU extraction queues. Each of these queue is then mapped to one of two CPU Extraction Groups. For each extraction group there is a scheduler (strict or round robin) which selects between the CPU extraction queues mapped to the same group.

When injecting frames, there are two CPU Injection Groups available where for instance one can be used for the Frame DMA and one can be used for manually injected frames. A scheduler (Strict or round robin) selects between the two injection groups meaning the switch core only sees one stream of frames being injected.

## 6.8.1 Forwarding to CPU

Several mechanisms can be used to trigger redirection or copying of frames to the CPU. They are listed in the following table.

*Table 189 •* **Configurations for Redirecting or Copying Frames to the CPU**

| Frame Type | Configuration (Including Selection of Extraction Queue) | Redirect or Copy |
|---|---|---|
| IEEE 802.1D Reserved Range<br>DMAC = 01-80-C2-00-00-0x | ANA:PORT:CPU_FWD_BPDU_CFG<br>ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL | Redirect |
| IEEE 802.1D Allbridge<br>DMAC = 01-80-C2-00-00-10 | ANA:PORT:<br>CPU_FWD_CFG.CPU_ALLBRIDGE_REDIR_ENA<br>ANA::CPUQ_CFG.CPUQ_ALLBRIDGE | Redirect |
| IEEE 802.1D GARP Range<br>DMAC = 01-80-C2-00-00-2x | ANA:PORT:CPU_FWD_GARP_CFG<br>ANA::CPUQ_8021_CFG.CPUQ_GARP_VAL | Redirect |
| IEEE 802.1D CCM/Link Trace Range<br>DMAC = 01-80-C2-00-00-3x | ANA:PORT:CPU_FWD_CCM_CFG<br>ANA::CPUQ_8021_CFG.CPUQ_CCM_VAL | Redirect |
| IGMP (IPv4) | ANA:PORT:CPU_IGMP_REDIR_ENA<br>ANA::CPUQ_CFG.CPUQ_IGMP | Redirect |
| IP Multicast Control (IPv4) | ANA:PORT:CPU_IPMC_CTRL_COPY_ENA<br>ANA::CPUQ_CFG.CPUQ_IPMC_CTRL | Copy |
| MLD (IPv6) | ANA:PORT:CPU_MLD_REDIR_ENA<br>ANA::CPUQ_CFG.CPUQ_MLD | Redirect |
| CPU-based learning | ANA:PORT:PORT_CFG.LEARNCPU<br>ANA::CPUQ_CFG.CPUQ_LRN | Copy |

*Table 189 •* **Configurations for Redirecting or Copying Frames to the CPU *(continued)***

| Frame Type | Configuration (Including Selection of Extraction Queue) | Redirect or Copy |
|---|---|---|
| CPU-based learning of locked MAC table entries seen on a new port | ANA:PORT:<br>PORT_CFG.LOCKED_PORTMOVE_CPU<br>ANA::CPUQ_CFG.CPUQ_LOCKED_PORTMOVE | |
| CPU-based learning of frames exceeding learn limit in MAC table | ANA:PORT:PORT_CFG.LIMIT_CPU<br>ANA::CPUQ_CFG.CPUQ_LRN | |
| MAC table match using MAC table | ANA::MACACCESS.MAC_CPU_COPY<br>ANA::CPUQ_CFG.CPUQ_MAC_COPY | Copy |
| MAC table match using PGID table | ANA::MACACCESS.DEST_IDX<br>ANA::PGID.PGID (bit 26)<br>ANA::PGID.CPUQ_DST_PGID | Redirect or copy |
| Flooded frames | ANA::MACACCESS.DEST_IDX<br>ANA::PGID.PGID (bit 26)<br>ANA::PGID.CPUQ_DST_PGID | Redirect or copy |
| Any frame received on selected ports | ANA:PORT:CPU_SRC_COPY_ENA<br>ANA:CPUQ_CFG.CPUQ_SRC_COPY | Copy |
| Mirroring | ANA::MIRRORPORTS (bit 26)<br>ANA::CPUQ_CFG:CPUQ_MIRROR<br>For more information about mirroring, see Mirroring, page 220. | Copy |
| VCAP IS2 rules | For more information about IS2, see VCAP IS2, page 67. | Redirect or copy |
| SFlow | ANA::CPUQ_CFG.CPUQ_SFLOW<br>For more information about SFlow, see sFlow Sampling, page 100. | Copy |

## 6.8.2 Frame Extraction

The CPU receives frames through the eight CPU extraction queues in the CPU port module. The eight queues are using resources (memory and frame descriptor pointers) from the shared queue system and are subject to the thresholds and congestion rules programmed for the CPU port (port 26) and the shared queue system in general.

The CPU can read frames from the CPU extraction queues in two ways:

- Reading registers in the CPU port module. For more information, see Frame Extraction, page 121.
- FDMA from CPU port module to RAM. For more information, see Frame DMA, page 150.

The switch core may place the eight-byte long CPU extraction header before the DMAC or after the SMAC (REW::PORT_CFG.IFH_INSERT_MODE). The CPU extraction header contains relevant side band information about the frame such as the frame's classification result (VLAN tag information, DSCP, or QoS class) and the reason for sending the frame to the CPU. For more information about the contents of the CPU extraction header, see CPU Extraction Header, page 121.

## 6.8.3 Frame Injection

The CPU can inject frames through the two CPU injection groups. The two groups merge into one injection queue through the injection scheduler (DEVCPU_QS::INJ_GRP_CFG). The injection queue uses resources (memory and frame descriptor pointers) from the shared queue system and is subject to the thresholds and congestion rules programmed for the CPU port (port 26) and the shared queue system in general.

The CPU can write frames to the CPU injection groups in two ways:

- Registers access to the CPU port module. For more information, see CPU Extraction and Injection, page 235.
- FDMA to CPU port module. For more information, see Frame DMA, page 150.

The first eight bytes of a frame written into a CPU group is an injection header containing relevant side band information about how the frame must be processed by the switch core. For more information, see Table 97, page 123.

### 6.8.4 Frame Extraction and Injection Using An External CPU

The following table lists the configuration registers associated with using an external CPU.

***Table 190 •*** **Configuration Registers When Using An External CPU**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| SYS::EXT_CPU_CFG.EXT_CPU_PORT | Port number where external CPU is connected. | None |
| SYS::EXT_CPU_CFG.EXT_CPUQ_MSK | Configures which CPU Extraction Queues are sent to the external CPU. | None |
| REW::PORT_CFG.IFH_INSERT_ENA | Enables the insertion of the CPU extraction header in egress frames. | Per port |
| REW::PORT_CFG.IFH_INSERT_MODE | Controls the position of the CPU extraction header. | Per port |
| SYS::PORT_MODE.INCL_INJ_HDR | Enables ingress port to look for CPU injection header in incoming frames. | Per port |

An external CPU can connect up to any front port module and use the Ethernet interface for extracting and injecting frames into the switch core.

**Note** If an external CPU is connected by means of the serial interface or parallel interface, the frame extraction and injection is performed. For more information, see Frame Extraction, page 237 and Frame Injection, page 237.

When extracting frames, the CPU extraction header can be placed before the DMAC (in the preamble) or after the SMAC (REW::PORT_CFG.IFH_INSERT_MODE). For more information about the contents of the eight-byte long extraction header, see Frame Extraction, page 237.

When injecting frames, the CPU injection header controls whether a frame is processed by the analyzer or forwarded directly to the destination set specified in the injection header. The injection header must be placed before destination MAC address in the frame. For more information about the contents of the eight-byte long injection header, see Frame Injection, page 237.

An internal and external CPU may coexist in a dual CPU system where the two CPUs handles different run-time protocols. When extracting CPU frames, it is selectable which CPU extraction queues are connected to the external CPU and which remain connected to the internal CPU (SYS::EXT_CPU_CFG.EXT_CPUQ_MSK). If a frame is forwarded to the CPU for more than one reason (for example, a BPDU which is also a learn frame), the frame can be forwarded to both the internal CPU extraction queues and to the external CPU.

## 6.9 Audio Video Bridging

Audio Video Bridging (AVB) defined by the IEEE 802.1 Audio/Video Bridging Task Group enables the delivery of time-synchronized, low-latency audio and video streaming services through Ethernet networks.

In an audio/video network it must be possible to synchronize multiple streams in time so that playback is rendered correctly. For example, keeping audio and video of a movie synchronized or keeping audio for multiple speakers in phase.

To guarantee consistent delivery of the streaming services, it must be possible to reserve network resources while the application needs it. For the switching equipment in the network, this means allocating enough bandwidth to support the streaming and to configure QoS handling so that latency is within the boundaries specified by the application.

Additionally, the worst-case delay through the network must be low and preferably deterministic so that an AVB system appears responsive to user interaction. A delay also has significant impact on the buffering requirement in the source and destination equipment.

The devices support all aspects of AVB, such as:

- Precise time synchronization defined by IEEE 802.1AS. This is a standard for synchronizing time in all participating nodes.The standard specifies the use of IEEE 1588 in the context of a VLAN-aware LAN switch. For more information about time synchronization, see Hardware Timestamping for AVB, page 124.
- Traffic shaping and scheduling of streaming services defined by IEEE 802.1Qav. Traffic shaping reduces bursting of data, and scheduling ensures that allocated bandwidth requirements are met. The devices implement eight queues per egress port, with shaping per queue and per port. The scheduler allows queues 6 and 7 to be strict while queues 0 through 5 are weighted. This ensures that time-sensitive data enqueued in queue 6 or 7 can be served before best-effort traffic enqueued in queue 5 or less. The shaper implements a non-bursty transmission mode so that the transmission times for AVB frames are evenly spread out. This reduces the effect of AVB frames being bunched together while reducing buffer requirements in destination equipment. For more information about the shaper and scheduler implementation, see Scheduler and Shaper, page 112.
- Admission control and resource allocation defined by IEEE 802.1Qav. The Stream Reservation Protocol (SRP) relies on the MMRP and MSRP. signaling protocols SRP frames can be redirected to the CPU using the GARP MAC address filter in the switch core.

## 6.10 Energy Efficient Ethernet

Defined by IEEE 802.3az, Energy Efficient Ethernet (EEE) provides a mechanism for reducing the energy consumption on Ethernet links during times of low utilization. Basically, when the transmission queues on a link are empty, the connecting macros and PHYs can be put into a sleep mode using Low-Power Idles (LPI), where the energy consumption is reduced by turning off unused circuits. When data is ready again for transmission, the macros and PHYs are waked up and data can flow again. The reaction time for bringing the link alive again is in the range of microseconds, so no data is lost due to low-power idles, however, data will experience increased latency.

Both internal PHYs and internal SerDes macros support EEE in both the Rx and Tx direction.

The following table lists configuration registers related to using Energy Efficient Ethernet.

***Table 191 •*** **Configuration Registers When Using Energy Efficient Ethernet**

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| SYS:PORT:EEE_CFG | Queue system configuration of EEE. | Per port |
| SYS::EEE_THRESH | EEE thresholds used by queue system. | None |
| PORT::PCS1G_LPI_CFG | Low power idle configuration for the PCS. | Per SerDes port |
| PORT::PCS1G_LPI_WAKE_ERROR_CNT | Wake error counter. | Per SerDes port |
| PORT::PCS1G_LPI_STATUS | Low power idle status. | Per SerDes port |

*Table 191 •* **Configuration Registers When Using Energy Efficient Ethernet *(continued)***

| Register/Register Field | Description/Value | Replication |
|---|---|---|
| HSIO::SERDES1G_MISC_CFG | Enable LPI in 1G SerDes. | Per SerDes port |
| HSIO::SERDES6G_MISC_CFG | Enable LPI in 6G SerDes. | Per SerDes port |
| IEEE Clause 45 PHY registers | EEE configuration for the internal PHYs. | Per Copper PHY port |

Ports with internal copper PHYs support LPI for 100BASE-TX and 1000BASE-T and can also reduce the transmit signal amplitude in a 10BASE-Te mode.

For ports with SerDes, the PCS supports LPI for all modes. When the PCS is in LPI, the connecting SerDes macro is also in LPI.

To enable Energy Efficient Ethernet, configure the following functions:

- Enable the ports for EEE and configure the timers and thresholds in the queue system to determine when the system will attempt to enter the LPI state and how fast it can wake up again.

Enable LPI for the relevant ports in PCS, SerDes macros, and internal PHYs. For more information, see PCS, page 18, SERDES1G, page 22, SERDES6G, page 26, and Cat5 Twisted Pair Media Interface, page 33.

# 7 Registers

This section provides information about the programming interface, register maps, register descriptions, and register tables of the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

In writing to registers with reserved bits, use a read-modify-write technique, where the entire register is read, but only the user bits to be changed are modified. Do not change the values of registers and bits marked as reserved. Their read state should not be considered static or unchanging. Unspecified registers and bits must be written to 0 and can be ignored when read.

## 7.1 Targets and Base Addresses

The following table lists all register targets and associated base addresses for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices. The next level lists registers groups and offsets within targets, and the deepest level lists registers within the register groups.

Both register groups and registers may be replicated (repeated) a number of times. The repeat-count and the distance between two repetitions is listed in the “Instances and Address Spacing” column of the tables. If there is only one instance, the spacing is omitted. The “Offset within Target”/”Offset within Register Group” columns hold the offset of the first instance of the register group/register.

***Table 192 •*** **List of Targets and Base Addresses**

| Target Name | Base Address | Description | Details |
|---|---|---|---|
| DEVCPU_ORG | 0x60000000 | CPU Device Origin | Page 242 |
| SYS | 0x60010000 | Switching Engine Configuration | Page 245 |
| ANA | 0x60020000 | Analyzer Configuration | Page 272 |
| REW | 0x60030000 | Rewriter Configuration | Page 304 |
| ES0 | 0x60040000 | VCAP ES0 Configuration | Page 308 |
| IS1 | 0x60050000 | VCAP IS1 Configuration | Page 308 |
| IS2 | 0x60060000 | VCAP IS2 Configuration | Page 308 |
| DEVCPU_GCB | 0x60070000 | CPU Device General Configuration | Page 342 |
| DEVCPU_QS | 0x60080000 | CPU Device Queue System | Page 381 |
| DEVCPU_PI | 0x60090000 | CPU Device Parallel Interface | Page 388 |
| HSIO | 0x600A0000 | High Speed I/O SerDes Configuration | Page 392 |
| DEV[0] | 0x601E0000 | Port Configuration (GMII) | Page 412 |
| DEV[1] | 0x601F0000 | Port Configuration (GMII) | Page 412 |
| DEV[2] | 0x60200000 | Port Configuration (GMII) | Page 412 |
| DEV[3] | 0x60210000 | Port Configuration (GMII) | Page 412 |
| DEV[4] | 0x60220000 | Port Configuration (GMII) | Page 412 |
| DEV[5] | 0x60230000 | Port Configuration (GMII) | Page 412 |
| DEV[6] | 0x60240000 | Port Configuration (GMII) | Page 412 |
| DEV[7] | 0x60250000 | Port Configuration (GMII) | Page 412 |
| DEV[8] | 0x60260000 | Port Configuration (GMII) | Page 412 |
| DEV[9] | 0x60270000 | Port Configuration (GMII) | Page 412 |
| DEV[10] | 0x60280000 | Port Configuration (GMII/SERDES) | Page 421 |
| DEV[11] | 0x60290000 | Port Configuration (GMII/SERDES) | Page 421 |

*Table 192 •* **List of Targets and Base Addresses *(continued)***

| Target Name | Base Address | Description | Details |
|---|---|---|---|
| DEV[12] | 0x602A0000 | Port Configuration (SERDES) | Page 421 |
| DEV[13] | 0x602B0000 | Port Configuration (SERDES) | Page 421 |
| DEV[14] | 0x602C0000 | Port Configuration (SERDES) | Page 421 |
| DEV[15] | 0x602D0000 | Port Configuration (SERDES) | Page 421 |
| DEV[16] | 0x602E0000 | Port Configuration (SERDES) | Page 421 |
| DEV[17] | 0x602F0000 | Port Configuration (SERDES) | Page 421 |
| DEV[18] | 0x60300000 | Port Configuration (SERDES) | Page 421 |
| DEV[19] | 0x60310000 | Port Configuration (SERDES) | Page 421 |
| DEV[20] | 0x60320000 | Port Configuration (SERDES) | Page 421 |
| DEV[21] | 0x60330000 | Port Configuration (SERDES) | Page 421 |
| DEV[22] | 0x60340000 | Port Configuration (SERDES) | Page 421 |
| DEV[23] | 0x60350000 | Port Configuration (SERDES) | Page 421 |
| DEV[24] | 0x60360000 | Port Configuration (SERDES) | Page 421 |
| DEV[25] | 0x60370000 | Port Configuration (SERDES) | Page 421 |
| ICPU_CFG | 0x70000000 | VCore Configuration | Page 443 |
| UART | 0x70100000 | VCore UART Configuration | Page 505 |
| TWI | 0x70100400 | VCore Two-Wire Interface Configuration | Page 517 |
| SBA | 0x70110000 | VCore Shared Bus Arbiter Configuration | Page 540 |
| GPDMA | 0x70110800 | VCore GPDMA Configuration | Page 543 |
| PHY | MIIM | PHY Configuration | Page 563 |

## 7.2 DEVCPU_ORG

*Table 193 •* **Register Groups in DEVCPU_ORG**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ORG | 0x00000000 | 1 | Origin registers | Page 242 |

### 7.2.1 DEVCPU_ORG:ORG

**Parent:** DEVCPU_ORG

**Instances:** 1

*Table 194 •* **Registers in ORG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ERR_ACCESS_DROP | 0x00000000 | 1 | Target Module ID is Unknown | Page 243 |

*Table 194 •* **Registers in ORG *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ERR_TGT | 0x00000008 | 1 | Target Module is Busy | Page 243 |
| ERR_CNTS | 0x0000000C | 1 | Error Counters | Page 244 |
| CFG_STATUS | 0x0000001C | 1 | Configuration and Status Register | Page 244 |

### 7.2.1.1 DEVCPU_ORG:ORG:ERR_ACCESS_DROP

**Parent:** DEVCPU_ORG:ORG

**Instances:** 1

*Table 195 •* **Fields in ERR_ACCESS_DROP**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NO_ACTION_STICKY | 24 | Sticky | Sticky bit that - when set - indicates that at least one request was received by a target, but the target did not do anything with it (Eg. access to a non existing register)<br>'0': No errors occurred.<br>'1': At least one request was received with no action. | 0x0 |
| TGT_MODULE_NO_ACTION_STICKY | 23:16 | R/O | Target Module ID.<br>When the sticky_no_action bit is set, this field holds the ID of the last target that received a request that didn't resolve in an action.<br>0x01 : Module id 1<br>0xFF : module id 255 | 0x00 |
| UTM_STICKY | 8 | Sticky | Sticky bit that - when set - indicates that at least one request for an unknown target module has been done.<br>'0': No errors occurred.<br>'1': At least one request to an unknown target has been done. | 0x0 |
| TGT_MODULE_UTM_STICKY | 7:0 | R/O | Target Module ID.<br>When the sticky_utm bit is set, this field holds the ID of the last target that was unknown.<br>0x01 : Module id 1<br>0xFF : module id 255 | 0x00 |

### 7.2.1.2 DEVCPU_ORG:ORG:ERR_TGT

**Parent:** DEVCPU_ORG:ORG

**Instances:** 1

Write all ones to this register to clear it.

*Table 196 •* **Fields in ERR_TGT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BSY_STICKY | 8 | Sticky | Sticky bit that - when set - indicates that at least one request was not processed because the target was busy.<br>'0': No error has occurred<br>'1': A least one request was dropped due to that the target was busy. | 0x0 |
| TGT_MODULE_BSY | 7:0 | R/O | Target Module ID.<br>When the sticky_bsy bit is set, this field holds the ID of the last target that was unable to process a request.<br>0x01 : Module id 1<br>0xFF : Module id 255 | 0x00 |

### 7.2.1.3 DEVCPU_ORG:ORG:ERR_CNTS

**Parent:** DEVCPU_ORG:ORG

**Instances:** 1

*Table 197 •* **Fields in ERR_CNTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NO_ACTION_CNT | 31:24 | R/W | No action Counter.<br>Counts the number of requests that were not processed by the Target Module, because the target did not know what to do ( e.g. access to a non-existing register ).<br>This counter saturates at max. | 0x00 |
| UTM_CNT | 23:16 | R/W | Unknown Target Counter.<br>Counts the number of requests that were not processed by the Target Module, because the target was no found.<br>This counter saturates at max. | 0x00 |
| BUSY_CNT | 15:8 | R/W | Busy Counter.<br>Counts the number of requests that were not processed by the Target Module, because it was busy. This may be because the Target Module was waiting for access to/from its host.<br>This counter saturates at max. | 0x00 |

### 7.2.1.4 DEVCPU_ORG:ORG:CFG_STATUS

**Parent:** DEVCPU_ORG:ORG

**Instances:** 1

*Table 198 •* **Fields in CFG_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RD_ERR_STICKY | 1 | Sticky | If a new read access is initialized before the previous read access has completed this sticky bit is set. Both the 1st and 2nd read access will be handled, but the 2nd access will overwrite data from the 1st access.<br>'0': A read access that has been initialized before the previous read access had completed has never occurred.<br>'1': At least one time a read access has been initialized before the previous read access had completed. | 0x0 |
| ACCESS_IN_PROGRESS | 0 | R/O | When set a access is in progress.<br>'0': No access is in progress.<br>'1': A access is in progress. | 0x0 |

## 7.3 SYS

*Table 199 •* **Register Groups in SYS**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SYSTEM | 0x000081B0 | 1 | Switch Configuration | Page 246 |
| SCH | 0x0000845C | 1 | Scheduler registers | Page 253 |
| SCH_LB | 0x00003800 | 1 | Scheduler leaky bucket registers | Page 257 |
| RES_CTRL | 0x00004000 | 1024<br>0x00000008 | Watermarks and status for egress queue system | Page 259 |
| PAUSE_CFG | 0x000085A4 | 1 | Watermarks for egress queue system | Page 261 |
| MMGT | 0x000037A0 | 1 | Memory manager status | Page 263 |
| MISC | 0x000037AC | 1 | Miscellaneous | Page 264 |
| STAT | 0x00000000 | 3558<br>0x00000004 | Frame statistics | Page 264 |
| PTP | 0x00008688 | 1 | Precision time protocol | Page 265 |
| POL | 0x00006000 | 256<br>0x00000020 | General policer configuration | Page 267 |
| POL_MISC | 0x00008704 | 1 | Flow control configuration | Page 269 |
| ISHP | 0x00008000 | 27<br>0x00000010 | Ingress shaper configuration | Page 270 |

## 7.3.1 SYS:SYSTEM

**Parent:** SYS

**Instances:** 1

*Table 200* • **Registers in SYSTEM**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| RESET_CFG | 0x00000000 | 1 | Core reset control | Page 246 |
| VLAN_ETYPE_CFG | 0x00000008 | 1 | S-tag Ethernet Type | Page 247 |
| PORT_MODE | 0x0000000C | 28<br>0x00000004 | Per device port configuration | Page 247 |
| FRONT_PORT_MODE | 0x0000007C | 26<br>0x00000004 | Various Ethernet port configurations | Page 248 |
| SWITCH_PORT_MODE | 0x000000E4 | 27<br>0x00000004 | Various switch port mode settings | Page 248 |
| FRM_AGING | 0x00000150 | 1 | Configure Frame Aging | Page 248 |
| STAT_CFG | 0x00000154 | 1 | Statistics configuration | Page 249 |
| EEE_CFG | 0x00000158 | 26<br>0x00000004 | Control Energy Efficient Ethernet operation per front port. | Page 250 |
| EEE_THRES | 0x000001C0 | 1 | Thresholds for delayed EEE queues | Page 251 |
| IGR_NO_SHARING | 0x000001C4 | 1 | Control shared memory users | Page 251 |
| EGR_NO_SHARING | 0x000001C8 | 1 | Control shared memory users | Page 252 |
| SW_STATUS | 0x000001CC | 27<br>0x00000004 | Various status info per switch port | Page 252 |
| EQ_TRUNCATE | 0x00000238 | 27<br>0x00000004 | Truncate frames in queue | Page 252 |
| EQ_PREFER_SRC | 0x000002A4 | 1 | Precedence for source ports | Page 252 |
| EXT_CPU_CFG | 0x000002A8 | 1 | External CPU port configuration | Page 253 |

### 7.3.1.1 SYS:SYSTEM:RESET_CFG

**Parent:** SYS:SYSTEM

**Instances:** 1

Controls reset and initialization of the switching core. Proper startup sequence is:

- Enable memories

- Initialize memories

- Enable core

*Table 201 •* **Fields in RESET_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CORE_ENA | 2 | R/W | Switch core is enabled when this field is set. | 0x0 |
| MEM_ENA | 1 | R/W | Core memory controllers are enabled when this field is set. | 0x0 |
| MEM_INIT | 0 | One-shot | Initialize core memories. Field is automatically cleared when operation is complete ( approx. 40 us). | 0x0 |

### 7.3.1.2 SYS:SYSTEM:VLAN_ETYPE_CFG

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 202 •* **Fields in VLAN_ETYPE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_S_TAG_ETYPE_VAL | 15:0 | R/W | Custom Ethernet Type for S-tags. Tags with TPID = 0x88A8 are always recognized as S-tags. | 0x88A8 |

### 7.3.1.3 SYS:SYSTEM:PORT_MODE

**Parent:** SYS:SYSTEM

**Instances:** 28

These configurations exists per frontport and for each of the two CPU ports (26+27).

*Table 203 •* **Fields in PORT_MODE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 4:3 | R/W | Must be set to its default. | 0x2 |
| L3_PARSE_CFG | 2 | R/W | Enable frame analysis on Layer-3 and Layer-4 protocol information. If cleared, all frames are seen as non-IP and are handled accordingly. This affects all blocks using IP information such as classification, TCAM lookups, IP flooding and forwarding, and DSCP rewriting. | 0x1 |
| DEQUEUE_DIS | 1 | R/W | Disable dequeuing from the egress queues. Frames are not discarded, but may become aged when dequeuing is re-enabled. | 0x0 |

*Table 203 •* **Fields in PORT_MODE *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INCL_INJ_HDR | 0 | R/W | Enable parsing of 64-bit injection header, which must be prepended all frames received on this port. | 0x0 |

### 7.3.1.4 SYS:SYSTEM:FRONT_PORT_MODE

**Parent:** SYS:SYSTEM

**Instances:** 26

*Table 204 •* **Fields in FRONT_PORT_MODE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| HDX_MODE | 0 | R/W | Enables the queue system to support the half duplex mode. Must be set for a port when enabled for half-duplex mode (MAC_MODE_ENA.FDX_ENA cleared). | 0x0 |

### 7.3.1.5 SYS:SYSTEM:SWITCH_PORT_MODE

**Parent:** SYS:SYSTEM

**Instances:** 27

*Table 205 •* **Fields in SWITCH_PORT_MODE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PORT_ENA | 3 | R/W | Enable port for any frame transfer. Frames to or from a port with PORT_ENA cleared are discarded. | 0x0 |
| RESERVED | 2 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 1 | R/W | Must be set to its default. | 0x1 |

### 7.3.1.6 SYS:SYSTEM:FRM_AGING

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 206 •* **Fields in FRM_AGING**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAX_AGE | 31:0 | R/W | Frames are aged and removed from the queue system when the frame's age timer becomes two. The frame age timer is increased for all frames whenever the configured time, MAX_AGE, has passed. The unit is 4 ns. Effectively, this means that a frame is aged when the frame has waited in the queue system between one or two times the period specified by MAX_AGE.<br><br>A value of zero disables the aging. A value less than 6000 (24 us) is illegal. | 0x00000000 |

### 7.3.1.7 SYS:SYSTEM:STAT_CFG

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 207 •* **Fields in STAT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_GREEN_CNT_MODE | 10 | R/W | Counter mode for the Tx priority counters for green frames (CNT_TX_GREEN_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |
| TX_YELLOW_CNT_MODE | 9 | R/W | Counter mode for the Tx priority counters for green frames (CNT_TX_YELLOW_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |
| DROP_GREEN_CNT_MODE | 8 | R/W | Counter mode for the drop counters for green frames (CNT_DR_GREEN_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |
| DROP_YELLOW_CNT_MODE | 7 | R/W | Counter mode for the drop counters for green frames (CNT_DR_YELLOW_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |
| STAT_CLEAR_PORT | 5:1 | R/W | Select which port to clear counters for. | 0x00 |

*Table 207 •* **Fields in STAT_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STAT_CLEAR_SHOT | 0 | One-shot | Set STAT_CLEAR_SHOT to clear all counters for the port selected by STAT_CLEAR_PORT port. Auto-cleared when complete (1us). | 0x0 |

### 7.3.1.8 SYS:SYSTEM:EEE_CFG

**Parent:** SYS:SYSTEM

**Instances:** 26

*Table 208 •* **Fields in EEE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_ENA | 29 | R/W | Enable EEE operation on the port.<br><br>A port enters the low power mode when no egress queues have data ready.<br><br>The port is activated when one of the following conditions is true:<br>- A queue has been non-empty for EEE_TIMER_AGE.<br>- A queue has more than EEE_HIGH_FRAMES frames pending.<br>- A queue has more than EEE_HIGH_BYTES bytes pending.<br>- A queue is marked as a fast queue, and has data pending. | 0x0 |
| EEE_FAST_QUEUES | 28:21 | R/W | Queues set in this mask activate the egress port immediately when any of the queues have data available. | 0x00 |
| EEE_TIMER_AGE | 20:14 | R/W | Maximum time frames in any queue must wait before the port is activated. The default value corresponds to 48 us.<br><br>Time = 4**(EEE_TIMER_AGE/16) * (EEE_TIMER_AGE mod 16) microseconds | 0x23 |

*Table 208 •* **Fields in EEE_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_TIMER_WAKEUP | 13:7 | R/W | Time from the egress port is activated until frame transmission is restarted. Default value corresponds to 16 us.<br>Time = 4**(EEE_TIMER_WAKEUP/16) * (EEE_TIMER_WAKEUP mod 16) microseconds | 0x14 |
| EEE_TIMER_HOLDOFF | 6:0 | R/W | When all queues are empty, the port is kept active until this time has passed. Default value corresponds to 5 us.<br>Time = 4**(EEE_TIMER_HOLDOFF/16) * (EEE_TIMER_HOLDOFF mod 16) microseconds | 0x05 |

### 7.3.1.9 SYS:SYSTEM:EEE_THRES

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 209 •* **Fields in EEE_THRES**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_HIGH_BYTES | 15:8 | R/W | Maximum number of bytes in a queue before egress port is activated. Unit is 48 bytes. | 0x00 |
| EEE_HIGH_FRAMES | 7:0 | R/W | Maximum number of frames in a queue before the egress port is activated. Unit is 1 frame. | 0x00 |

### 7.3.1.10 SYS:SYSTEM:IGR_NO_SHARING

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 210 •* **Fields in IGR_NO_SHARING**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IGR_NO_SHARING | 26:0 | R/W | Control whether frames received on the port may use shared resources. If ingress port or queue has reserved memory left to use, frame enqueuing is always allowed.<br>0: Use shared memory as well<br>1: Do not use shared memory | 0x0000000 |

### 7.3.1.11 SYS:SYSTEM:EGR_NO_SHARING

**Parent:** SYS:SYSTEM

**Instances:** 1

***Table 211 •*** **Fields in EGR_NO_SHARING**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EGR_NO_SHARING | 26:0 | R/W | Control whether frames forwarded to the port may use shared resources. If egress port or queue has reserved memory left to use, frame enqueuing is always allowed.<br>0: Use shared memory as well<br>1: Do not use shared memory | 0x0000000 |

### 7.3.1.12 SYS:SYSTEM:SW_STATUS

**Parent:** SYS:SYSTEM

**Instances:** 27

***Table 212 •*** **Fields in SW_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EQ_AVAIL | 9:2 | R/O | Status bit per egress queue indicating whether data is ready for transmission. | 0x00 |
| PORT_LPI | 1 | R/O | Status bit indicating whether port is in low-power-idle due to the LPI algorithm (EEE_CFG). If set, transmissions are held back. | 0x0 |
| PORT_RX_PAUSED | 0 | R/O | Status bit indicating whether the switch core is instructing the MAC to pause the ingress port. | 0x0 |

### 7.3.1.13 SYS:SYSTEM:EQ_TRUNCATE

**Parent:** SYS:SYSTEM

**Instances:** 27

***Table 213 •*** **Fields in EQ_TRUNCATE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EQ_TRUNCATE | 7:0 | R/W | If a bit is set, frames transmitted from corresponding egress queue are truncated to 92 bytes. | 0x00 |

### 7.3.1.14 SYS:SYSTEM:EQ_PREFER_SRC

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 214* • **Fields in EQ_PREFER_SRC**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EQ_PREFER_SRC | 26:0 | R/W | When multiple sources have data in the same priority, ingress ports set in this mask are preferred over ingress ports not set when arbitrating frames from ingress to egress. When multiple ports are set, the arbitration between these ports are round-robin. | 0x4000000 |

#### 7.3.1.15 SYS:SYSTEM:EXT_CPU_CFG

**Parent:** SYS:SYSTEM

**Instances:** 1

*Table 215* • **Fields in EXT_CPU_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_CPU_PORT | 12:8 | R/W | Select the port to use as the external CPU port. | 0x1B |
| EXT_CPUQ_MSK | 7:0 | R/W | Frames destined for a CPU extraction queue set in this mask are sent to the external CPU defined by EXT_CPU_PORT instead of the internal CPU. | 0x00 |

### 7.3.2 SYS:SCH

**Parent:** SYS

**Instances:** 1

*Table 216* • **Registers in SCH**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| LB_DWRR_FRM_ADJ | 0x00000000 | 1 | Leaky bucket frame adjustment | Page 254 |
| LB_DWRR_CFG | 0x00000004 | 26<br>0x00000004 | Leaky bucked frame adjustment | Page 254 |
| SCH_DWRR_CFG | 0x0000006C | 26<br>0x00000004 | Deficit weighted round robin control register | Page 254 |
| SCH_SHAPING_CTRL | 0x000000D8 | 26<br>0x00000004 | Scheduler shaping control register | Page 255 |
| SCH_LB_CTRL | 0x00000140 | 1 | Leaky bucket control | Page 256 |

*Table 216 •* **Registers in SCH *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SCH_CPU | 0x00000144 | 1 | Map CPU queues to CPU ports | Page 257 |

### 7.3.2.1 SYS:SCH:LB_DWRR_FRM_ADJ

**Parent:** SYS:SCH

**Instances:** 1

*Table 217 •* **Fields in LB_DWRR_FRM_ADJ**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FRM_ADJ | 4:0 | R/W | Value added to leaky buckets and DWRR each time a frame is scheduled. If set to 20, this corresponds to inclusion of minimum Ethernet IFG and preamble.<br><br>0-31: Number of bytes added at start of frame | 0x00 |

### 7.3.2.2 SYS:SCH:LB_DWRR_CFG

**Parent:** SYS:SCH

**Instances:** 26

*Table 218 •* **Fields in LB_DWRR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FRM_ADJ_ENA | 0 | R/W | If enabled, the value configured in SCH_LB_DWRR_FRM_ADJ.FRM_ADJ is added to the frame length for each frame.<br><br>The modified frame length is used by both the leaky bucket and DWRR algorithm.<br>0:Disable frame length adjustment.<br>1:Enable frame length adjustment. | 0x0 |

### 7.3.2.3 SYS:SCH:SCH_DWRR_CFG

**Parent:** SYS:SCH

**Instances:** 26

*Table 219 •* **Fields in SCH_DWRR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DWRR_MODE | 30 | R/W | Configure DWRR scheduling for port. Weighted- and strict prioritization can be configured.<br>0: All priorities are scheduled strict<br>1: The two highest priorities (6, 7) are strict. The rest is DWRR | 0x0 |
| COST_CFG | 29:0 | R/W | Queue cost configuration. Bit vector used to configure the cost of each priority.<br>Bits 4:0: Cost for queue 0.<br>Bits 9:5: Cost for queue 1.<br>Bits 14:10: Cost for queue 2.<br>Bits 19:15: Cost for queue 3.<br>Bits 24:20: Cost for queue 4.<br>Bits 29:25: Cost for queue 5.<br>Within each cost field, the following encoding is used:<br>0: Cost 1<br>1: Cost 2<br>...<br>31: Cost 32 | 0x00000000 |

### 7.3.2.4 SYS:SCH:SCH_SHAPING_CTRL

**Parent:** SYS:SCH

**Instances:** 26

*Table 220 •* **Fields in SCH_SHAPING_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PRIO_SHAPING_ENA | 7:0 | R/W | Enable priority shaping. If enabled the BW of a priority is limited to SCH_LB::LB_RATE.<br>xxxxxxx1: Enable shaping for Prio 0<br>xxxxxx1x: Enable shaping for Prio 1<br>...<br>1xxxxxxx: Enable shaping for Prio N | 0x00 |
| PORT_SHAPING_ENA | 8 | R/W | Enable port shaping. If enabled the total BW of a port is limited to SCH_LB::LB_RATE.<br>0: Disable port shaping<br>1: Enable port shaping | 0x0 |

*Table 220 •* **Fields in SCH_SHAPING_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PRIO_LB_EXS_ENA | 23:16 | R/W | Allow this queue to use excess bandwidth. If none of the priorities are allowed (by their priority LB) to transmit.<br><br>The resulting BW of a queue is a function of the port- and queue LBs, the DWRR and the excess enable bit:<br>1) Port LB closed. Hold back frames.<br>2) Port LB open -> Use strict- or DWRR scheduling to distribute traffic between open Queue LBs<br>3) All Queue LBs closed -> Hold back frames except for Queues which have PRIO_LB_EXS_ENA set. The excess BW is distributed using strict- or DWRR scheduling.<br><br>xxxxxxx1: Enable excess BW for Prio 0<br>xxxxxx1x: Enable excess BW for Prio 1<br>...<br>1xxxxxxx: Enable excess BW for Prio N | 0x00 |
| PRIO_LB_AVB_ENA | 31:24 | R/W | Enable AV Bridging (AVB) shaping mode for queues. In AVB mode the burst capacity of a queue is limited. An AVB queue can only build up burst capacity when it has traffic to send.<br><br>xxxxxxx1: Enable AVB mode for Prio 0<br>xxxxxx1x: Enable AVB mode for Prio 1<br>...<br>1xxxxxxx: Enable AVB mode for Prio N | 0x00 |

### 7.3.2.5 SYS:SCH:SCH_LB_CTRL

**Parent:** SYS:SCH

**Instances:** 1

*Table 221 •* **Fields in SCH_LB_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LB_INIT | 0 | One-shot | Set to 1 to force a complete initialization of state and configuration of leaky buckets. Must be done before the scheduler is used. Field is automatically cleared whether initialization is complete.<br><br>0: No Action<br>1: Force initialization. | 0x0 |

#### 7.3.2.6 SYS:SCH:SCH_CPU

**Parent:** SYS:SCH

**Instances:** 1

*Table 222 •* **Fields in SCH_CPU**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SCH_CPU_MAP | 9:2 | R/W | Maps the 8 CPU queues to CPU port 26 or 27. Bit <n> set directs CPU queue <n> to CPU port 26/27. | 0x00 |
| SCH_CPU_RR | 1:0 | R/W | Set the scheduler for CPU port <n> to run round robin between queues instead of strict. | 0x0 |

### 7.3.3 SYS:SCH_LB

**Parent:** SYS

**Instances:** 1

Ethernet leaky bucket configuration per port and per priority.

The address of the configuration is based on the following layout: (Assume the priority count is 8)

0: Leaky bucket for priority 0 of port 0

1: Leaky bucket for priority 1 of port 0

2: Leaky bucket for priority 2 of port 0

3: Leaky bucket for priority 3 of port 0

4: Leaky bucket for priority 4 of port 0

5: Leaky bucket for priority 5 of port 0

6: Leaky bucket for priority 6 of port 0

7: Leaky bucket for priority 7 of port 0

8: Leaky bucket port 0

9: Leaky bucket for priority 0 of port 1

10: Leaky bucket for priority 1 of port 1

.

.

The configuration for each leaky bucket includes rate and threshold configuration.

***Table 223 •*** **Registers in SCH_LB**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| LB_THRES | 0x00000000 | 234<br>0x00000004 | Leaky bucket threshold | Page 258 |
| LB_RATE | 0x00000400 | 234<br>0x00000004 | Leaky bucket rate | Page 258 |

### 7.3.3.1 SYS:SCH_LB:LB_THRES

**Parent:** SYS:SCH_LB

**Instances:** 234

***Table 224 •*** **Fields in LB_THRES**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LB_THRES | 5:0 | R/W | Burst capacity of leaky buckets<br><br>The unit is 4KB (1KB = 1024Bytes). The largest supported threshold is 252KB when the register value is set to all "1"s.<br><br>Queue shaper Q on port P uses shaper 9*P+Q. Port shaper on port P uses shaper 9*P+8.<br>0: Always closed<br>1: Burst capacity = 4096 bytes<br>...<br>n: Burst capacity = n x 4096 bytes | 0x00 |

### 7.3.3.2 SYS:SCH_LB:LB_RATE

**Parent:** SYS:SCH_LB

**Instances:** 234

*Table 225 •* **Fields in LB_RATE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LB_RATE | 14:0 | R/W | Leaky bucket rate in unit of 100160 bps.<br><br>Queue shaper Q on port P uses shaper 9*P+Q. Port shaper on port P uses shaper 9*P+8.<br>0: Open until burst capacity is used, then closed.<br>1: Rate = 100160 bps<br>n: Rate = n x 100160 bps | 0x0000 |

## 7.3.4 SYS:RES_CTRL

**Parent:** SYS

**Instances:** 1024

*Table 226 •* **Registers in RES_CTRL**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| RES_CFG | 0x00000000 | 1 | Watermark configuration | Page 259 |
| RES_STAT | 0x00000004 | 1 | Resource status | Page 260 |

### 7.3.4.1 SYS:RES_CTRL:RES_CFG

**Parent:** SYS:RES_CTRL

**Instances:** 1

The queue system tracks four resource consumptions:

Resource 0: Memory tracked per source

Resource 1: Frame references tracked per source

Resource 2: Memory tracked per destination

Resource 3: Frame references tracked per destination

Before a frame is added to the queue system, some conditions must be met:

- Reserved memory for the specific (SRC, PRIO) or for the specific SRC is available

OR

- Reserved memory for the specific (DST,PRIO) or for the specific DST is available

OR

- Shared memory is available

The frame reference resources are checked for availability like the memory resources. Enqueuing of a frame is allowed if both the memory resource check and the frame reference resource check succeed.

The extra resources consumed when enqueuing a frame are first taken from the reserved (SRC,PRIO), next from the reserved SRC, and last from the shared memory area. The same is done for DST. Both memory consumptions and frame reference consumptions are updated.

The register is layed out the following way:

Index 0-215: Reserved amount for (x,PRIO) at index 8*x+PRIO, x=SRC or DST

Index 224-250: Reserved amount for (x)

Resource 0 is accessed at index 0-255, 1 at index 256-511 etc.

The amount of shared memory is located at index 255. An extra watermark at 254 is used for limiting amount of shared memory used before yellow traffic is discarded.

The amount of shared references is located at index 511. An extra watermark at 510 is used for limiting amount of shared references for yellow traffic.

At index 216-223 there is a watermarks per priority used for limiting how much of the shared buffer must be used per priority.

Likewise at offset 472 there are priority watermarks for references.

The allocation size for memory tracking is 48 bytes, and all frames is added a 4 byte header internally.

***Table 227 •*** **Fields in RES_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WM_HIGH | 10:0 | R/W | Watermark for resource.<br>Note, the default value depends on the index. Refer to the congestion scheme documentation for details.<br>Bit 10: Unit; 0:1, 1:16<br>Bits 9-0: Value to be multiplied with unit | 0x000 |

### 7.3.4.2 SYS:RES_CTRL:RES_STAT

**Parent:** SYS:RES_CTRL

**Instances:** 1

***Table 228 •*** **Fields in RES_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INUSE | 27:14 | R/W | Current consumption for corresponding watermark in RES_CFG. | 0x0000 |

*Table 228 •* **Fields in RES_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAXUSE | 13:0 | R/W | Maximum consumption for corresponding watermark in RES_CFG. | 0x0000 |

## 7.3.5 SYS:PAUSE_CFG

**Parent:** SYS

**Instances:** 1

*Table 229 •* **Registers in PAUSE_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PAUSE_CFG | 0x00000000 | 27<br>0x00000004 | Watermarks for flow control condition per switch port. | Page 261 |
| PAUSE_TOT_CFG | 0x0000006C | 1 | Configure total memory pause condition | Page 262 |
| ATOP | 0x00000070 | 27<br>0x00000004 | Tail dropping level | Page 262 |
| ATOP_TOT_CFG | 0x000000DC | 1 | Total raw memory use before tail dropping is activated | Page 262 |
| EGR_DROP_FORCE | 0x000000E0 | 1 | Configures egress ports for flowcontrol | Page 263 |

### 7.3.5.1 SYS:PAUSE_CFG:PAUSE_CFG

**Parent:** SYS:PAUSE_CFG

**Instances:** 27

*Table 230 •* **Fields in PAUSE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAUSE_START | 22:12 | R/W | Start pausing ingress stream when the amount of memory consumed by the port exceeds this watermark. The TOTPAUSE condition must also be met.<br>See RES_CFG | 0x7FF |
| PAUSE_STOP | 11:1 | R/W | Stop pausing ingress stream when the amount of memory consumed by the port is below this watermark.<br>See RES_CFG. | 0x7FF |

*Table 230 •* **Fields in PAUSE_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAUSE_ENA | 0 | R/W | Enable pause feedback to the MAC, allowing transmission of pause frames or HDX collisions to limit ingress data rate. | 0x0 |

### 7.3.5.2 SYS:PAUSE_CFG:PAUSE_TOT_CFG

**Parent:** SYS:PAUSE_CFG

**Instances:** 1

*Table 231 •* **Fields in PAUSE_TOT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAUSE_TOT_START | 21:11 | R/W | Assert TOTPAUSE condition when total memory allocation is above this watermark.<br>See RES_CFG | 0x000 |
| PAUSE_TOT_STOP | 10:0 | R/W | Deassert TOTPAUSE condition when total memory allocation is below this watermark.<br>See RES_CFG | 0x000 |

### 7.3.5.3 SYS:PAUSE_CFG:ATOP

**Parent:** SYS:PAUSE_CFG

**Instances:** 27

*Table 232 •* **Fields in ATOP**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ATOP | 10:0 | R/W | When a source port consumes more than this level in the packet memory, frames are tail dropped, unconditionally of destination.<br>See RES_CFG | 0x7FF |

### 7.3.5.4 SYS:PAUSE_CFG:ATOP_TOT_CFG

**Parent:** SYS:PAUSE_CFG

**Instances:** 1

*Table 233 •* **Fields in ATOP_TOT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ATOP_TOT | 10:0 | R/W | Tail dropping is activate on a port when the port use has exceeded the ATOP watermark for the port, and the total memory use has exceeded this watermark.<br>See RES_CFG | 0x7FF |

#### 7.3.5.5 SYS:PAUSE_CFG:EGR_DROP_FORCE

**Parent:** SYS:PAUSE_CFG

**Instances:** 1

*Table 234 •* **Fields in EGR_DROP_FORCE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EGRESS_DROP_FORCE | 26:0 | R/W | When enabled for a port, frames to the port are discarded, even when the ingress port is enabled for flow control. Applicable to egress ports that should not create head-of-line blocking in ingress ports operating in flow control mode. An example is the CPU port. | 0x0000000 |

### 7.3.6 SYS:MMGT

**Parent:** SYS

**Instances:** 1

*Table 235 •* **Registers in MMGT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MMGT | 0x00000000 | 1 | Packet Memory Status | Page 263 |
| EQ_CTRL | 0x00000008 | 1 | Egress queue status | Page 264 |

#### 7.3.6.1 SYS:MMGT:MMGT

**Parent:** SYS:MMGT

**Instances:** 1

*Table 236* • **Fields in MMGT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FREECNT | 19:8 | R/O | Number of 192-byte free memory words. | 0x000 |

### 7.3.6.2 SYS:MMGT:EQ_CTRL

**Parent:** SYS:MMGT

**Instances:** 1

*Table 237* • **Fields in EQ_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FP_FREE_CNT | 12:0 | R/O | Number of free frame references. | 0x0000 |

## 7.3.7 SYS:MISC

**Parent:** SYS

**Instances:** 1

*Table 238* • **Registers in MISC**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| REPEATER | 0x00000018 | 1 | Frame repeating setup | Page 264 |

### 7.3.7.1 SYS:MISC:REPEATER

**Parent:** SYS:MISC

**Instances:** 1

*Table 239* • **Fields in REPEATER**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| REPEATER | 26:0 | R/W | A bit set in this mask makes the corresponding port skip dequeing from the queue selected by the scheduler. This can be used for simple frame generation and scheduler experiments. | 0x0000000 |

## 7.3.8 SYS:STAT

**Parent:** SYS

**Instances:** 3558

These registers are used for accessing all frame statistics.

*Table 240 •* **Registers in STAT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CNT | 0x00000000 | 1 | Counter values | Page 265 |

### 7.3.8.1 SYS:STAT:CNT

**Parent:** SYS:STAT

**Instances:** 1

*Table 241 •* **Fields in CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT | 31:0 | R/W | Counter values.<br><br>The counters are layed in three main blocks where each port has a share within the block:<br>Rx counters: 0x000 - 0x488<br>- port0: 0x000 - 0x02A<br>- port1: 0x02B - 0x055<br>...<br>- port26 (CPU): 0x45E - 0x488<br><br>Tx counters: 0x800 - 0xB44<br>- port0: 0x800 - 0x81E<br>- port1: 0x81F - 0x83D<br>...<br>- port26 (CPU): 0xB26 - 0xB44<br><br>Drop counters: 0xC00 - 0xDE5<br>- port0: 0xC00 - 0xC11<br>- port1: 0xC12 - 0xC23<br>...<br>- port26 (CPU): 0xDD4 - 0xDE5<br><br>SYS::STAT_CFG and ANA::AGENCTRL control whether bytes or frames are counted for specific counters. Counters are cleared through SYS::STAT_CFG. | 0x00000000 |

## 7.3.9 SYS:PTP

**Parent:** SYS

**Instances:** 1

*Table 242* • **Registers in PTP**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PTP_STATUS | 0x00000000 | 1 | Stored timestamp and timestamp queue status | Page 266 |
| PTP_DELAY | 0x00000004 | 1 | Timestamp value | Page 266 |
| PTP_CFG | 0x00000008 | 28<br>0x00000004 | Configuration of Rx and Tx hardware timestamping | Page 267 |
| PTP_NXT | 0x00000078 | 1 | Advancing the timestamp queue | Page 267 |

### 7.3.9.1 SYS:PTP:PTP_STATUS

**Parent:** SYS:PTP

**Instances:** 1

*Table 243* • **Fields in PTP_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_OVFL | 12 | R/O | If set, the timestamp queue has overflown implying a timestamp entry could not enqueued. The PTP_OVFL bit is not cleared until the timestamp queue is completely empty. | 0x0 |
| PTP_MESS_VLD | 11 | R/O | A timestamp entry is ready for reading. PTP_MESS_ID, PTP_MESS_TXPORT, and PTP_DELAY contain the data of the timestamp entry. | 0x0 |
| PTP_MESS_ID | 10:5 | R/O | Timestamp identifier for head-of-line timestamp entry. | 0x00 |
| PTP_MESS_TXPORT | 4:0 | R/O | The transmit port for the head-of-line timestamp entry. | 0x00 |

### 7.3.9.2 SYS:PTP:PTP_DELAY

**Parent:** SYS:PTP

**Instances:** 1

*Table 244 •* **Fields in PTP_DELAY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_DELAY | 31:0 | R/O | The timestamp value for the head-of-line timestamp entry. The timestamp value is the frame's arrival time if the transmit port is the CPU port. Otherwise the timestamp value is the frame's residence time. Unit is 4 ns. | 0x00000000 |

### 7.3.9.3 SYS:PTP:PTP_CFG

**Parent:** SYS:PTP

**Instances:** 28

*Table 245 •* **Fields in PTP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_1STEP_DIS | 17 | R/W | Disable updating of the correction field in PTP frames. This overrules the IS2 PTP_ENA[0] action. | 0x0 |
| PTP_2STEP_DIS | 16 | R/W | Disable adding the entries to the timestamp queue. This overrules the IS2 PTP_ENA[1] action. | 0x0 |
| IO_TX_DELAY | 15:8 | R/W | Delay added to the sampled departure time. Unit is 4 ns. | 0x00 |
| IO_RX_DELAY | 7:0 | R/W | Delay subtracted from the sampled arrival time. Unit is 4 ns. | 0x00 |

### 7.3.9.4 SYS:PTP:PTP_NXT

**Parent:** SYS:PTP

**Instances:** 1

*Table 246 •* **Fields in PTP_NXT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_NXT | 0 | One-shot | Advance to the next timestamp entry. Registers PTP_STATUS and PTP_DELAY points to the next entry. | 0x0 |

## 7.3.10 SYS:POL

**Parent:** SYS

**Instances:** 256

General purpose policers selected by port configuration and ACL actions

*Table 247* • **Registers in POL**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| POL_PIR_CFG | 0x00000000 | 1 | Peak Information Rate configuration for this policer | Page 268 |
| POL_MODE_CFG | 0x00000008 | 1 | Common configuration for this policer | Page 268 |
| POL_PIR_STATE | 0x0000000C | 1 | State of this policer | Page 269 |

### 7.3.10.1 SYS:POL:POL_PIR_CFG

**Parent:** SYS:POL

**Instances:** 1

*Table 248* • **Fields in POL_PIR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PIR_RATE | 20:6 | R/W | Accepted rate for this policer. Unit is 100 kbps. | 0x0000 |
| PIR_BURST | 5:0 | R/W | Burst capacity of this policer. Unit is 4 kilobytes. | 0x00 |

### 7.3.10.2 SYS:POL:POL_MODE_CFG

**Parent:** SYS:POL

**Instances:** 1

*Table 249* • **Fields in POL_MODE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IPG_SIZE | 9:5 | R/W | Size of IPG to add to each frame if line rate policing is chosen in FRM_MODE. | 0x14 |
| FRM_MODE | 4:3 | R/W | Accounting mode of this policer.<br>0: Line rate. Police bytes including IPG_SIZE.<br>1: Data rate. Police bytes excluding IPG.<br>2. Frame rate. Police frames with rate unit = 100 fps and burst unit = 32.8 frames.<br>3: Frame rate. Police frame with rate unit = 1 fps and burst unit = 0.3 frames. | 0x0 |

*Table 249 •* **Fields in POL_MODE_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OVERSHOOT_ENA | 0 | R/W | If set, overshoot is allowed. This implies that a frame of any length is accepted if the policer is open even if the frame causes the bucket to use more than the remaining capacity.<br>If cleared, overshoot is not allowed. This implies that it is checked that the frame will not use more than the remaining capacity in the bucket before accepting the frame. | 0x1 |

### 7.3.10.3 SYS:POL:POL_PIR_STATE

**Parent:** SYS:POL

**Instances:** 1

*Table 250 •* **Fields in POL_PIR_STATE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PIR_LVL | 21:0 | R/W | Current fill level of this policer. Unit is 0.5 bits. | 0x000000 |

## 7.3.11 SYS:POL_MISC

**Parent:** SYS

**Instances:** 1

*Table 251 •* **Registers in POL_MISC**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| POL_FLOWC | 0x00000000 | 27<br>0x00000004 | Flow control configuration per policer | Page 269 |
| POL_HYST | 0x0000006C | 1 | Set delay between flow control clearings | Page 270 |

### 7.3.11.1 SYS:POL_MISC:POL_FLOWC

**Parent:** SYS:POL_MISC

**Instances:** 27

*Table 252 •* **Fields in POL_FLOWC**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| POL_FLOWC | 0 | R/W | Use MAC flow control for lowering ingress rate<br>0: Standard policing. Frames are discarded when the rate is exceeded.<br>1: Flow control policing. Policer instructs the MAC to issue pause frames when the rate is exceeded. | 0x0 |

### 7.3.11.2 SYS:POL_MISC:POL_HYST

**Parent:** SYS:POL_MISC

**Instances:** 1

*Table 253 •* **Fields in POL_HYST**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| POL_FC_HYST | 9:4 | R/W | Set hysteresis for when to re-open a bucket after the burst capacity has been used. Unit is 1 kilobytes. This applies to policer in flow control mode (POL_FLOWC=1). | 0x02 |
| POL_DROP_HYST | 3:0 | R/W | Set hysteresis for when to re-open a bucket after the burst capacity has been used. Unit is 2 kilobytes. This applies to policer in drop mode (POL_FLOWC=0). | 0x0 |

## 7.3.12 SYS:ISHP

**Parent:** SYS

**Instances:** 27

*Table 254 •* **Registers in ISHP**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ISHP_CFG | 0x00000000 | 1 | Rate and burst configuration | Page 270 |
| ISHP_MODE_CFG | 0x00000004 | 1 | Mode of operation | Page 271 |
| ISHP_STATE | 0x00000008 | 1 | State of this shaper | Page 271 |

### 7.3.12.1 SYS:ISHP:ISHP_CFG

**Parent:** SYS:ISHP

**Instances:** 1

***Table 255 •*** **Fields in ISHP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ISHP_RATE | 21:7 | R/W | Accepted rate for this shaper. Unit is 100 kbps. | 0x0000 |
| ISHP_BURST | 6:1 | R/W | Burst capacity of this shaper. Unit is 4kB | 0x00 |
| ISHP_ENA | 0 | R/W | Enable ingress shaping for this port. | 0x0 |

### 7.3.12.2 SYS:ISHP:ISHP_MODE_CFG

**Parent:** SYS:ISHP

**Instances:** 1

***Table 256 •*** **Fields in ISHP_MODE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ISHP_IPG_SIZE | 6:2 | R/W | Size of IPG to add each frame if line rate shaping is chosen in ISHP_MODE. | 0x14 |
| ISHP_MODE | 1:0 | R/W | Accounting mode of this shaper.<br>0: Line rate. Shape bytes including IPG_size<br>1: Data rate. Shape bytes excluding IPG<br>2. Frame rate. Shape frames with rate unit = 100 fps and burst unit = 32.8 frames.<br>3: Frame rate. Shape frame with rate unit = 1 fps and burst unit = 0.3 frames. | 0x0 |

### 7.3.12.3 SYS:ISHP:ISHP_STATE

**Parent:** SYS:ISHP

**Instances:** 1

***Table 257 •*** **Fields in ISHP_STATE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ISHP_LVL | 21:0 | R/W | Current fill level of this shaper. Unit is 0.5 bits. | 0x000000 |

# 7.4 ANA

*Table 258* • **Register Groups in ANA**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ANA | 0x00000D80 | 1 | General analyzer configuration | Page 272 |
| ANA_TABLES | 0x00001000 | 1 | MAC, VLAN, and PGID table configuration | Page 282 |
| PORT | 0x00000000 | 27<br>0x00000080 | Per port configurations for Classifier | Page 289 |
| COMMON | 0x00000E38 | 1 | Common configurations for Classifier | Page 299 |

## 7.4.1 ANA:ANA

**Parent:** ANA

**Instances:** 1

*Table 259* • **Registers in ANA**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ADVLEARN | 0x00000000 | 1 | Advanced Learning Setup | Page 273 |
| VLANMASK | 0x00000004 | 1 | VLAN Source Port Mask | Page 273 |
| ANAGEFIL | 0x00000008 | 1 | Aging Filter | Page 273 |
| ANEVENTS | 0x0000000C | 1 | Event Sticky Bits | Page 274 |
| STORMLIMIT_BURST | 0x00000010 | 1 | Storm policer burst | Page 276 |
| STORMLIMIT_CFG | 0x00000014 | 4<br>0x00000004 | Storm Policer configuration | Page 276 |
| ISOLATED_PORTS | 0x00000024 | 1 | Private VLAN Mask for isolated ports | Page 276 |
| COMMUNITY_PORTS | 0x00000028 | 1 | Private VLAN Mask for community ports | Page 277 |
| AUTOAGE | 0x0000002C | 1 | Auto Age Timer | Page 277 |
| MACTOPTIONS | 0x00000030 | 1 | MAC Table Options | Page 278 |
| LEARNDISC | 0x00000034 | 1 | Learn Discard Counter | Page 278 |
| AGENCTRL | 0x00000038 | 1 | Analyzer Configuration | Page 279 |
| MIRRORPORTS | 0x0000003C | 1 | Mirror Target Ports | Page 280 |
| EMIRRORPORTS | 0x00000040 | 1 | Egress Mirror Mask | Page 280 |
| FLOODING | 0x00000044 | 1 | Standard flooding configuration | Page 281 |

*Table 259 •* **Registers in ANA *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| FLOODING_IPMC | 0x00000048 | 1 | Flooding configuration for IP multicasts | Page 281 |
| SFLOW_CFG | 0x0000004C | 27<br>0x00000004 | SFlow sampling configuration per port | Page 281 |

### 7.4.1.1 ANA:ANA:ADVLEARN

**Parent:** ANA:ANA

**Instances:** 1

*Table 260 •* **Fields in ADVLEARN**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_CHK | 26 | R/W | If this bit is set, a frame discarded because of VLAN ingress filtering is not subject to learning. VLAN ingress filtering is controlled by the VLAN_SRC_CHK flag in the VLAN table (see VLANACCESS register) or the VLANMASK register. | 0x0 |
| LEARN_MIRROR | 25:0 | R/W | Learn frames are also forwarded to ports marked in this mask. | 0x0000000 |

### 7.4.1.2 ANA:ANA:VLANMASK

**Parent:** ANA:ANA

**Instances:** 1

*Table 261 •* **Fields in VLANMASK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLANMASK | 26:0 | R/W | Mask for requiring VLAN ingress filtering. If the bit for the frame's physical ingress port is set in this mask, then the port must be member of ingress frame's VLAN (VLANACCESS.VLAN_PORT_MASK), otherwise the frame is discarded. | 0x0000000 |

### 7.4.1.3 ANA:ANA:ANAGEFIL

**Parent:** ANA:ANA

**Instances:** 1

This register sets up which entries are touched by an aging operation (manual as well as automatic aging).

In this way, it is possible to have different aging periods in each VLAN and to have quick removal of entries on specific ports.

The register also affects the GET_NEXT MAC table command. When using the register to control the behavior of GET_NEXT, it is recommended to disable automatic aging while executing the GET_NEXT command.

***Table 262 •*** **Fields in ANAGEFIL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AGE_LOCKED | 19 | R/W | Select entries to age. If cleared, unlocked entries will be aged and potentially removed. If set, locked entries will be aged but not removed. | 0x0 |
| PID_EN | 18 | R/W | If set, only MAC table entries with a destination index matching PID_VAL are aged. | 0x0 |
| PID_VAL | 17:13 | R/W | Destination index used in selective aging. | 0x00 |
| VID_EN | 12 | R/W | If set, only MAC table entries with a VID matching VID_VAL are aged. | 0x0 |
| VID_VAL | 11:0 | R/W | VID used in selective aging. | 0x000 |

### 7.4.1.4 ANA:ANA:ANEVENTS

**Parent:** ANA:ANA

**Instances:** 1

***Table 263 •*** **Fields in ANEVENTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AUTOAGE | 24 | Sticky | An AUTOAGE run was performed. | 0x0 |
| STORM_DROP | 22 | Sticky | A frame was discarded, because it exceeded the flooding storm limitations configured in STORMLIMIT. | 0x0 |
| LEARN_DROP | 21 | Sticky | A frame was discarded, because it was subject to learning, and the DropMode flag was set in ADVLEARN. | 0x0 |
| AGED_ENTRY | 20 | Sticky | An entry was removed at CPU Learn, or CPU requested an aging process. | 0x0 |
| CPU_LEARN_FAILED | 19 | Sticky | A learn operation failed due to hash table depletion. CPU-based learning only. | 0x0 |

*Table 263 •* **Fields in ANEVENTS** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AUTO_LEARN_FAILED | 18 | Sticky | A learn operation of incoming source MAC address failed due to hash table depletion. Hardware-based learning only. | 0x0 |
| LEARN_REMOVE | 17 | Sticky | An entry was removed when learning a new source MAC address. | 0x0 |
| AUTO_LEARNED | 16 | Sticky | An entry was learned from an incoming frame. Hardware-based learning only. | 0x0 |
| AUTO_MOVED | 15 | Sticky | A station was moved to another port. | 0x0 |
| CLASSIFIED_DROP | 13 | Sticky | A frame was not forwarded due to classification (such as BPDUs). | 0x0 |
| CLASSIFIED_COPY | 12 | Sticky | A frame was copied to the CPU due to classification. | 0x0 |
| VLAN_DISCARD | 11 | Sticky | A frame was discarded due to lack of VLAN membership on source port. | 0x0 |
| FWD_DISCARD | 10 | Sticky | A frame was discarded due to missing forwarding state on source port. | 0x0 |
| MULTICAST_FLOOD | 9 | Sticky | A frame was flooded with multicast flooding mask. | 0x0 |
| UNICAST_FLOOD | 8 | Sticky | A frame was flooded with unicast flooding mask. | 0x0 |
| DEST_KNOWN | 7 | Sticky | A frame was forwarded with known destination MAC address. | 0x0 |
| BUCKET3_MATCH | 6 | Sticky | A destination was found in hash table bucket 3. | 0x0 |
| BUCKET2_MATCH | 5 | Sticky | A destination was found in hash table bucket 2. | 0x0 |
| BUCKET1_MATCH | 4 | Sticky | A destination was found in hash table bucket 1. | 0x0 |
| BUCKET0_MATCH | 3 | Sticky | A destination was found in hash table bucket 0. | 0x0 |
| CPU_OPERATION | 2 | Sticky | A CPU-initiated operation on the MAC or VLAN table was processed. Default is 1 due to auto-initialization of the MAC and VLAN table. | 0x1 |
| DMAC_LOOKUP | 1 | Sticky | A destination address was looked up in the MAC table. | 0x0 |
| SMAC_LOOKUP | 0 | Sticky | A source address was looked up in the MAC table. | 0x0 |

### 7.4.1.5 ANA:ANA:STORMLIMIT_BURST

**Parent:** ANA:ANA

**Instances:** 1

*Table 264* • **Fields in STORMLIMIT_BURST**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STORM_BURST | 3:0 | R/W | Allowed number of frames in a burst is 2**STORM_BURST. The maximum allowed burst is 4096 frames, which corresponds to STORM_BURST = 12. The STORM_BURST is common for all storm policers. | 0x0 |

### 7.4.1.6 ANA:ANA:STORMLIMIT_CFG

**Parent:** ANA:ANA

**Instances:** 4

0: UC storm policer

1: BC storm policer

2: MC policer

3: Learn policer

*Table 265* • **Fields in STORMLIMIT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STORM_RATE | 6:3 | R/W | Allowed rate of storm policer is 2**STORM_UNIT frames per second or kiloframes per second. See STORM_UNIT. The maximum allowed rate is 1024 kiloframes per second, which corresponds to STORM_RATE = 10 with STORM_UNIT set to 0. | 0x0 |
| STORM_UNIT | 2 | R/W | If set, the base unit for the storm policer is one frame per second. If cleared, the base unit is one kiloframe per second. | 0x0 |
| STORM_MODE | 1:0 | R/W | Mode of operation for storm policer.<br>0: Disabled.<br>1: Police CPU destination only.<br>2: Police front port destinations only.<br>3: Police both CPU and front port destinations. | 0x0 |

### 7.4.1.7 ANA:ANA:ISOLATED_PORTS

**Parent:** ANA:ANA

**Instances:** 1

*Table 266 •* **Fields in ISOLATED_PORTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ISOL_PORTS | 26:0 | R/W | This mask is used in private VLANs applications. Promiscuous and community ports must be set and isolated ports must be cleared.<br><br>For frames classified to a private VLAN (see the VLAN_PRIV_VLAN field in VLAN table), the resulting VLAN mask is calculated as follows:<br>- Frames received on a promiscuous port use the VLAN mask directly.<br>- Frames received on a community port use the VLAN mask AND'ed with the ISOL_PORTS.<br>- Frames received on a isolated port use the VLAN mask AND'ed with the COMM_PORTS AND'ed with the ISOL_PORTS.<br><br>For frames classified to a non-private VLAN, this mask is not used. | 0x7FFFFFF |

### 7.4.1.8 ANA:ANA:COMMUNITY_PORTS

**Parent:** ANA:ANA

**Instances:** 1

*Table 267 •* **Fields in COMMUNITY_PORTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| COMM_PORTS | 26:0 | R/W | This mask is used in private VLANs applications. Promiscuous and isolated ports must be set and community ports must be cleared.<br><br>See ISOLATED_PORTS.ISOL_PORTS for details. | 0x7FFFFFF |

### 7.4.1.9 ANA:ANA:AUTOAGE

**Parent:** ANA:ANA

**Instances:** 1

*Table 268 •* **Fields in AUTOAGE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AGE_FAST | 21 | R/W | Sets the unit of PERIOD to 8.2 us. PERIOD must be a minimum of 3 when using the FAST option. | 0x0 |
| AGE_PERIOD | 20:1 | R/W | Time in seconds between automatic aging of a MAC table entry. Setting AGE_PERIOD to zero effectively disables automatic aging. An inactive unlocked MAC table entry is aged after 2*AGE_PERIOD. | 0x00000 |
| AUTOAGE_LOCKED | 0 | R/W | Also set the AGED_FLAG bit on locked entries. They will not be removed. | 0x0 |

### 7.4.1.10 ANA:ANA:MACTOPTIONS

**Parent:** ANA:ANA

**Instances:** 1

*Table 269 •* **Fields in MACTOPTIONS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| REDUCED_TABLE | 1 | R/W | When set, the MAC table will be reduced 256 entries (64 hash-chains of 4) | 0x0 |
| SHADOW | 0 | R/W | Enable MAC table shadow registers. The SHADOW bit affects the behavior of the READ command in MACACCESS.MAC_TABLE_CMD : With the shadow bit set, reading bucket 0 causes the remaining 3 buckets in the row to be stored in "shadow registers". Following read accesses to bucket 1-3 return the content of the shadow registers. This is useful when reading a MAC table, which can change while being read. | 0x0 |

### 7.4.1.11 ANA:ANA:LEARNDISC

**Parent:** ANA:ANA

**Instances:** 1

The total number of MAC table entries that have been or would have been learned, but have been discarded due to a lack of storage space.

*Table 270* • **Fields in LEARNDISC**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LEARNDISC | 31:0 | R/W | Number of discarded learn requests due to MAC table overflow (collisions or MAC table entry limits). | 0x00000000 |

### 7.4.1.12 ANA:ANA:AGENCTRL

**Parent:** ANA:ANA

**Instances:** 1

*Table 271* • **Fields in AGENCTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FID_MASK | 23:12 | R/W | Mask used to enable shared learning among multiple VLANs. The FID value used in learning and MAC table lookup is calculated as: FID = VID and (not FID_MASK) By default, FID_MASK is set to all-zeros, corresponding to independent VLAN learning. In this case FID becomes identical to VID. | 0x000 |
| IGNORE_DMAC_FLAGS | 11 | R/W | Do not react to flags found in the DMAC entry or the corresponding flags for flooded frames (FLOOD_IGNORE_VLAN). | 0x0 |
| IGNORE_SMAC_FLAGS | 10 | R/W | Do not react to flags found in the SMAC entry. Note, the IGNORE_VLAN flag is not checked for SMAC entries. | 0x0 |
| FLOOD_SPECIAL | 9 | R/W | Flood frames using the lowest 27 bits of DMAC as destination port mask. This is only added for testing purposes. | 0x0 |
| FLOOD_IGNORE_VLAN | 8 | R/W | VLAN mask is not applied to flooded frames. | 0x0 |
| MIRROR_CPU | 7 | R/W | Frames destined for the CPU extraction queues are also forwarded to the port set configured in MIRRORPORTS. | 0x0 |
| LEARN_CPU_COPY | 6 | R/W | If set, auto-learned stations get the CPU_COPY flag set in the MAC table entry. | 0x0 |
| LEARN_SRC_KILL | 5 | R/W | If set, auto-learned stations get the SRC_KILL flag set in the MAC table entry. | 0x0 |

*Table 271 •* **Fields in AGENCTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LEARN_IGNORE_VLAN | 4 | R/W | If set, auto-learned stations get the IGNORE_VLAN flag set in the MAC table entry. | 0x0 |
| CPU_CPU_KILL_ENA | 3 | R/W | If set, CPU injected frames are never sent back to the CPU. | 0x1 |
| GREEN_COUNT_MODE | 2 | R/W | Counter mode for the Rx priority counters for green frames (CNT_RX_GREEN_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |
| YELLOW_COUNT_MODE | 1 | R/W | Counter mode for the Rx priority counters for yellow frames (CNT_RX_YELLOW_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |
| RED_COUNT_MODE | 0 | R/W | Counter mode for the Rx priority counters for red frames (CNT_RX_RED_PRIO_x)<br>0: Count octets<br>1: Count frames | 0x1 |

### 7.4.1.13 ANA:ANA:MIRRORPORTS

**Parent:** ANA:ANA

**Instances:** 1

*Table 272 •* **Fields in MIRRORPORTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIRRORPORTS | 26:0 | R/W | Ports set in this mask receive a mirror copy. If CPU is included in mask (bit 26 set), then the frame is copied to CPU extraction queue CPUQ_CFG.CPUQ_MIRROR. | 0x0000000 |

### 7.4.1.14 ANA:ANA:EMIRRORPORTS

**Parent:** ANA:ANA

**Instances:** 1

*Table 273 •* **Fields in EMIRRORPORTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EMIRRORPORTS | 26:0 | R/W | Frames forwarded to ports in this mask are mirrored to the port set configured in MIRRORPORTS (i.e. egress port mirroring). | 0x0000000 |

### 7.4.1.15 ANA:ANA:FLOODING

**Parent:** ANA:ANA

**Instances:** 1

***Table 274 •*** **Fields in FLOODING**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FLD_UNICAST | 17:12 | R/W | Set the PGID mask to use when flooding unknown unicast frames. | 0x3F |
| FLD_BROADCAST | 11:6 | R/W | Set the PGID mask to use when flooding unknown broadcast frames. | 0x3F |
| FLD_MULTICAST | 5:0 | R/W | Set the PGID mask to use when flooding unknown multicast frames (except IP multicasts). | 0x3F |

### 7.4.1.16 ANA:ANA:FLOODING_IPMC

**Parent:** ANA:ANA

**Instances:** 1

***Table 275 •*** **Fields in FLOODING_IPMC**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FLD_MC4_CTRL | 23:18 | R/W | Set the PGID mask to use when flooding unknown IPv4 Multicast Control frames. | 0x3F |
| FLD_MC4_DATA | 17:12 | R/W | Set the PGID mask to use when flooding unknown IPv4 Multicast Data frames. | 0x3F |
| FLD_MC6_CTRL | 11:6 | R/W | Set the PGID mask to use when flooding unknown IPv6 Multicast Control frames. | 0x3F |
| FLD_MC6_DATA | 5:0 | R/W | Set the PGID mask to use when flooding unknown IPv6 Multicast Data frames. | 0x3F |

### 7.4.1.17 ANA:ANA:SFLOW_CFG

**Parent:** ANA:ANA

**Instances:** 27

*Table 276 •* **Fields in SFLOW_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SF_RATE | 13:2 | R/W | Probability of a frame being SFLOW sampled. Unit is 1/4096. A value of 0 makes 1/4096 of the candidates being forwarded to the SFLOW CPU extraction queue. A values of 4095 makes all candidates being forwarded. | 0x000 |
| SF_SAMPLE_RX | 1 | R/W | Enable SFLOW sampling of frames received on this port. | 0x0 |
| SF_SAMPLE_TX | 0 | R/W | Enable SFLOW sampling of frames transmitted on this port. | 0x0 |

## 7.4.2 ANA:ANA_TABLES

**Parent:** ANA

**Instances:** 1

*Table 277 •* **Registers in ANA_TABLES**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ANMOVED | 0x000001AC | 1 | Station Move Logger | Page 282 |
| MACHDATA | 0x000001B0 | 1 | MAC Address High | Page 283 |
| MACLDATA | 0x000001B4 | 1 | MAC Address Low | Page 283 |
| MACACCESS | 0x000001B8 | 1 | MAC Table Command | Page 283 |
| MACTINDX | 0x000001BC | 1 | MAC Table Index | Page 285 |
| VLANACCESS | 0x000001C0 | 1 | VLAN Table Command | Page 286 |
| VLANTIDX | 0x000001C4 | 1 | VLAN Table Index | Page 287 |
| PGID | 0x00000000 | 107<br>0x00000004 | Port Group Identifiers | Page 287 |
| ENTRYLIM | 0x00000200 | 27<br>0x00000004 | MAC Table Entry Limits | Page 288 |
| PTP_ID_HIGH | 0x000001C8 | 1 | PTP Identifiers 63-32 | Page 289 |
| PTP_ID_LOW | 0x000001CC | 1 | PTP Identifiers 31-0 | Page 289 |

### 7.4.2.1 ANA:ANA_TABLES:ANMOVED

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 278 •* **Fields in ANMOVED**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ANMOVED | 26:0 | R/W | Sticky bit set when a station has been learned on a port while already learned on another port (i.e. port move). The register is cleared by writing 1 to the bits to be cleared. This mask can be used to detect topology problems in the network, where stations are learned on multiple ports repeatedly. If some bits in this register get asserted repeatedly, the ports can be shut down, or management warnings can be issued. | 0x0000000 |

### 7.4.2.2 ANA:ANA_TABLES:MACHDATA

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 279 •* **Fields in MACHDATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VID | 27:16 | R/W | VID used in MAC table operations through MACACCESS. For read operations, the VID value is returned in this field. | 0x000 |
| MACHDATA | 15:0 | R/W | Most significant 16 MAC address bits used in MAC table operations through MACACCESS. | 0x0000 |

### 7.4.2.3 ANA:ANA_TABLES:MACLDATA

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 280 •* **Fields in MACLDATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MACLDATA | 31:0 | R/W | Lower 32 MAC address bits used in MAC table operations through MACACCESS. | 0x00000000 |

### 7.4.2.4 ANA:ANA_TABLES:MACACCESS

**Parent:** ANA:ANA_TABLES

**Instances:** 1

This register is used for updating or reading the MAC table from the CPU.

The command (MAC_TABLE_CMD) selects between different operations and uses the following encoding:

000 - IDLE:

The previous operation has completed.

001 - LEARN:

Insert/learn new entry in MAC table. Position given by (MAC, VID) in MACHDATA and MACLDATA.

010 - FORGET:

Delete/unlearn entry given by (MAC, VID) in MACHDATA and MACLDATA.

Both locked and unlocked entries are deleted.

011 - AGE:

Start an age scan on the MAC table.

100 - GET_NEXT:

Get the smallest entry in the MAC table numerically larger than the (MAC, VID) specified in MACHDATA and MACLDATA. The VID and MAC are evaluated as a 60-bit number with the VID being most significant.

101 - INIT:

Table is initialized (completely cleared).

110 - READ:

The READ command is divided into two modes: Direct mode and indirect mode.

Direct mode (read):

With MACACCESS.VALID cleared, the entry pointed to by MACTINDX.INDEX (row) and MACTINDX.BUCKET (column) is read.

Indirect mode (lookup):

With MACACCESS.VALID set, the entry pointed to by (MAC, VID) in the MACHDATA and MACLDATA is read.

111 - WRITE

Write entry. Address of the entry is specified in MACTINDX.INDEX (row) and MACTINDX.BUCKET (column).

An existing entry (locked or unlocked) is overwritten.

The MAC_TABLE_CMD must be IDLE before a new command can be issued.

The AGE and CLEAR commands run for approximately 50 us. The other commands execute immediately.

The flags IGNORE_VLAN and MAC_CPU_COPY are ignored for DMAC lookup if AGENCTRL.IGNORE_DMAC_FLAGS is set.

The flags SRC_KILL and MAC_CPU_COPY are ignored for SMAC lookup if AGENCTRL.IGNORE_SMAC_FLAGS is set.

***Table 281 •*** **Fields in MACACCESS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IP6_MASK | 18:16 | R/W | Bits 24:22 in the destination port mask for IPv6 entries. | 0x0 |
| MAC_CPU_COPY | 15 | R/W | Frames matching this entry are copied to the CPU extraction queue CPUQ_CFG.CPUQ_MAC. Applies to both SMAC and DMAC lookup. | 0x0 |
| SRC_KILL | 14 | R/W | Frames matching this entry are discarded. Applies only to the SMAC lookup. For discarding frames based on the DMAC lookup a NULL PGID mask can be used. | 0x0 |
| IGNORE_VLAN | 13 | R/W | The VLAN mask is ignored for this destination. Applies only to DMAC lookup. | 0x0 |
| AGED_FLAG | 12 | R/W | This flag is set on every aging run. Entry is removed if flag is already set. The flag is cleared when the entry is target for a SMAC lookup. Locked entries will not be removed. Bit is for IPv6 Multicast used for port 25. | 0x0 |
| VALID | 11 | R/W | Entry is valid. | 0x0 |
| ENTRY_TYPE | 10:9 | R/W | Type of entry:<br>0: Normal entry eligible for aging<br>1: Locked entry. Entry will not be removed by aging<br>2: IPv4 Multicast entry. Full portset in mac record<br>3: IPv6 Multicast entry. Full portset in mac record | 0x0 |
| DEST_IDX | 8:3 | R/W | Index for the destination masks table (PGID). For unicasts, this is a number from 0-EXB_PORT_CNT_MINUS_ONE. | 0x00 |
| MAC_TABLE_CMD | 2:0 | R/W | MAC Table Command. See below. | 0x0 |

### 7.4.2.5 ANA:ANA_TABLES:MACTINDX

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 282 •* **Fields in MACTINDX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BUCKET | 12:11 | R/W | Selects one of the four MAC table entries in a row. The row is addressed with the INDEX field. | 0x0 |
| M_INDEX | 10:0 | R/W | The index selects one of the 2048 MAC table rows. Within a row the entry is addressed by the BUCKET field | 0x000 |

### 7.4.2.6 ANA:ANA_TABLES:VLANACCESS

**Parent:** ANA:ANA_TABLES

**Instances:** 1

The VLAN_TBL_CMD field of this register is used for updating and reading the VLAN table. The command (VLAN_TBL_CMD) selects between different operations and uses the following encoding:

00 - IDLE:

The previous operation has completed.

01 - READ:

The VLAN table entry set in VLANTIDX.INDEX is returned in VLANACCESS.VLAN_PORT_MASK and the VLAN flags in VLANTIDX.

10 - WRITE:

The VLAN table entry pointed to by VLANTIDX.INDEX is updated with VLANACCESS.VLAN_PORT_MASK and the VLAN flags in VLANTIDX.

11 - INIT:

The VLAN table is initialized to default values (all ports are members of all VLANs).

The VLAN_TBL_CMD must be IDLE before a new command can be issued. The INIT command run for approximately 50 us whereas the other commands execute immediately. When an operation has completed, VLAN_TBL_CMD changes to IDLE.

*Table 283 •* **Fields in VLANACCESS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_PORT_MASK | 28:2 | R/W | Frames classified to this VLAN can only be sent to ports in this mask. Note that the CPU port module is always member of all VLANs and its VLAN membership can therefore not be configured through this mask. | 0x3FFFFFF |

*Table 283 •* **Fields in VLANACCESS *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_TBL_CMD | 1:0 | R/W | VLAN Table Command. | 0x0 |

### 7.4.2.7 ANA:ANA_TABLES:VLANTIDX

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 284 •* **Fields in VLANTIDX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_PRIV_VLAN | 15 | R/W | If set, a VLAN is a private VLAN. See PRIV_VLAN_MASK for details. | 0x0 |
| VLAN_LEARN_DISABLED | 14 | R/W | Disable learning for this VLAN. | 0x0 |
| VLAN_MIRROR | 13 | R/W | If set, all frames classified to this VLAN are mirrored to the port set configured in MIRRORPORTS. | 0x0 |
| VLAN_SRC_CHK | 12 | R/W | If set, VLAN ingress filtering is enabled for this VLAN. If set, a frame's ingress port must be member of the frame's VLAN, otherwise the frame is discarded. | 0x0 |
| V_INDEX | 11:0 | R/W | Index used to select VLAN table entry for read/write operations (see VLANACCESS). This value equals the VID. | 0x000 |

### 7.4.2.8 ANA:ANA_TABLES:PGID

**Parent:** ANA:ANA_TABLES

**Instances:** 107

Three port masks are applied to all frames, allowing transmission to a port if the corresponding bit is set in all masks.

0-63: A mask is applied based on destination analysis

64-79: A mask is applied based on aggregation analysis

80-106: A mask is applied based on source port analysis

Destination analysis:

There are 64 destination masks in total. By default, the first 26 port masks only have the bit corresponding to their port number set. These masks should not be changed, except for aggregation.

The remaining destination masks are set to 0 by default and are available for use for Layer-2 multicasts and flooding (See FLOODING and FLOODING_IPMC).

Aggregation analysis:

The aggregation port masks are used to select only one port within each aggregation group. These 16 masks must be setup to select only one port in each aggregated port group.

For ports, which are not part of any aggregation group, the corresponding bits in all 16 masks must be set.

I.e. if no aggregation is configured, all masks must be set to all-ones.

The aggregation mask used for the forwarding of a given frame is selected by the frame's aggregation code (see AGGRCTRL).

Source port analysis:

The source port masks are used to prevent frames from being looped back to the ports on which they were received, and must be updated according to the

aggregation configuration. A frame that is received on port n, uses mask 80+n as a mask to filter out destination ports to avoid loopback, or to facilitate port grouping (port-based VLANs). The default values are that all bits are set except for the index number.

***Table 285 •*** **Fields in PGID**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PGID | 26:0 | R/W | When a mask is chosen, bit N must be set for the frame to be transmitted on port N. | 0x7FFFFFF |
| CPUQ_DST_PGID | 29:27 | R/W | CPU extraction queue used when CPU port is enabled in PGID. Only applicable for the destination analysis. | 0x0 |

### 7.4.2.9 ANA:ANA_TABLES:ENTRYLIM

**Parent:** ANA:ANA_TABLES

**Instances:** 27

***Table 286 •*** **Fields in ENTRYLIM**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRYLIM | 17:14 | R/W | Maximum number of unlocked entries in the MAC table learned on this port.<br>Locked entries and IPMC entries do not obey this limit.<br>Both auto-learned and unlocked CPU-learned entries obey this limit.<br>0: 1 entry<br>1: 2 entries<br>n: 2**n entries<br>>12: 8192 entries | 0xD |

*Table 286 •* **Fields in ENTRYLIM *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRYSTAT | 13:0 | R/W | Current number of unlocked MAC table entries learned on this port. | 0x0000 |

### 7.4.2.10 ANA:ANA_TABLES:PTP_ID_HIGH

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 287 •* **Fields in PTP_ID_HIGH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_ID_HIGH | 31:0 | R/W | Bit vector with current use of PTP timestamp identifiers 32 through 63. Timestamp identifier is 63 is reserved for signaling that no identifiers are available. A timestamp identifier is released by setting the corresponding bit.<br>Bit 0: Timestamp identifier 32<br>...<br>Bit 31: Timestamp identifier 63. | 0x00000000 |

### 7.4.2.11 ANA:ANA_TABLES:PTP_ID_LOW

**Parent:** ANA:ANA_TABLES

**Instances:** 1

*Table 288 •* **Fields in PTP_ID_LOW**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_ID_LOW | 31:0 | R/W | Bit vector with current use of PTP timestamp identifiers 0 through 31. A timestamp identifier is released by setting the corresponding bit.<br>Bit 0: Timestamp identifier 0<br>...<br>Bit 31: Timestamp identifier 31. | 0x00000000 |

## 7.4.3 ANA:PORT

**Parent:** ANA

**Instances:** 27

*Table 289 •* **Registers in PORT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VLAN_CFG | 0x00000000 | 1 | Port VLAN configuration | Page 290 |

*Table 289 •* **Registers in PORT *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| DROP_CFG | 0x00000004 | 1 | VLAN acceptance filtering | Page 291 |
| QOS_CFG | 0x00000008 | 1 | QoS and DSCP configuration | Page 292 |
| VCAP_CFG | 0x0000000C | 1 | VCAP configuration | Page 292 |
| QOS_PCP_DEI_MAP_CFG | 0x00000010 | 16 0x00000004 | Mapping of DEI and PCP to QoS class and drop precedence level | Page 294 |
| CPU_FWD_CFG | 0x00000050 | 1 | CPU forwarding of special protocols | Page 295 |
| CPU_FWD_BPDU_CFG | 0x00000054 | 1 | CPU forwarding of BPDU frames | Page 296 |
| CPU_FWD_GARP_CFG | 0x00000058 | 1 | CPU forwarding of GARP frames | Page 296 |
| CPU_FWD_CCM_CFG | 0x0000005C | 1 | CPU forwarding of CCM/Link trace frames | Page 296 |
| PORT_CFG | 0x00000060 | 1 | Special port configuration | Page 296 |
| POL_CFG | 0x00000064 | 1 | Policer selection | Page 298 |

### 7.4.3.1 ANA:PORT:VLAN_CFG

**Parent:** ANA:PORT

**Instances:** 1

*Table 290 •* **Fields in VLAN_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_AWARE_ENA | 20 | R/W | Enable VLAN awareness. If set, Q-tag headers are processed during the basic VLAN classification. If cleared, Q-tag headers are ignored during the basic VLAN classification. | 0x0 |
| VLAN_POP_CNT | 19:18 | R/W | Number of tag headers to remove from ingress frame.<br>0: Keep all tags.<br>1: Pop up to 1 tag (outer tag if available).<br>2: Pop up to 2 tags (outer and inner tag if available).<br>3: Reserved. | 0x0 |

*Table 290 •* **Fields in VLAN_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_INNER_TAG_ENA | 17 | R/W | Set if the inner Q-tag must be used instead of the outer Q-tag. If the received frame is single tagged, the outer tag is used. This bit influences the VLAN acceptance filter (DROP_CFG), the basic VLAN classification (VLAN_CFG), and the basic QoS classification (QOS_CFG). | 0x0 |
| VLAN_TAG_TYPE | 16 | R/W | Tag Protocol Identifier type for port-based VLAN.<br>0: C-tag (EtherType = 0x8100)<br>1: S-tag (EtherType = 0x88A8 or configurable value (VLAN_ETYPE_CFG)) | 0x0 |
| VLAN_DEI | 15 | R/W | DEI value for port-based VLAN. | 0x0 |
| VLAN_PCP | 14:12 | R/W | PCP value for port-based VLAN. | 0x0 |
| VLAN_VID | 11:0 | R/W | VID value for port-based VLAN. | 0x000 |

### 7.4.3.2 ANA:PORT:DROP_CFG

**Parent:** ANA:PORT

**Instances:** 1

*Table 291 •* **Fields in DROP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DROP_UNTAGGED_ENA | 6 | R/W | Drop untagged frames. | 0x0 |
| DROP_S_TAGGED_ENA | 5 | R/W | Drop S-tagged frames (VID different from 0 and EtherType = 0x88A8 or configurable value (VLAN_ETYPE_CFG)). | 0x0 |
| DROP_C_TAGGED_ENA | 4 | R/W | Drop C-tagged frames (VID different from 0 and EtherType = 0x8100). | 0x0 |
| DROP_PRIO_S_TAGGED_ENA | 3 | R/W | Drop S-tagged frames (VID=0 and EtherType = 0x88A8 or configurable value (VLAN_ETYPE_CFG)). | 0x0 |
| DROP_PRIO_C_TAGGED_ENA | 2 | R/W | Drop priority C-tagged frames (VID=0 and EtherType = 0x8100). | 0x0 |
| DROP_NULL_MAC_ENA | 1 | R/W | Drop frames with source or destination MAC address equal to 0x000000000000. | 0x0 |
| DROP_MC_SMAC_ENA | 0 | R/W | Drop frames with multicast source MAC address. | 0x0 |

### 7.4.3.3 ANA:PORT:QOS_CFG

**Parent:** ANA:PORT

**Instances:** 1

*Table 292* • **Fields in QOS_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DP_DEFAULT_VAL | 8 | R/W | Default drop precedence level. | 0x0 |
| QOS_DEFAULT_VAL | 7:5 | R/W | Default QoS class. | 0x0 |
| QOS_DSCP_ENA | 4 | R/W | If set, the DP level and QoS class can be based on DSCP values. | 0x0 |
| QOS_PCP_ENA | 3 | R/W | If set, DP level and QoS class can be based on the PCP and DEI bits for tagged frames. | 0x0 |
| DSCP_TRANSLATE_ENA | 2 | R/W | Set if the DSCP value must be translated before using the DSCP value. If set, the translated DSCP value is given from DSCP_CFG[DSCP].DSCP_TRANSLATE_VAL. | 0x0 |
| DSCP_REWR_CFG | 1:0 | R/W | Configure which DSCP values to rewrite based on DP level and QoS class. If the DSCP value is to be rewritten, then the new DSCP = DSCP_REWR_CFG[8*DP level + QoS class].DSCP_QOS_REWR_VAL.<br>0: Rewrite none.<br>1: Rewrite if DSCP=0<br>2: Rewrite for selected values configured in DSCP_CFG[DSCP].DSCP_REWR_ENA.<br>3: Rewrite all. | 0x0 |

### 7.4.3.4 ANA:PORT:VCAP_CFG

**Parent:** ANA:PORT

**Instances:** 1

*Table 293* • **Fields in VCAP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| S1_ENA | 29 | R/W | If S1 is enabled, each frame received on this port is processed and matched against the entries in the S1 TCAM. Each frame results in three lookups (two lookups to determine classification actions such as VLAN and QoS class, and one lookup to check host identity). | 0x0 |

*Table 293 •* **Fields in VCAP_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| S1_DMAC_DIP_ENA | 28:27 | R/W | Set if the destination MAC address and the destination IP address must be passed on to the S1 TCAM instead of the source MAC address and the source IP address. Bit 0 controls destination address information for first lookup in S1. Bit 1 controls destination address information for second lookup in S1. Note that the host identity lookup in S1 always uses source information. | 0x0 |
| S1_VLAN_INNER_TAG_ENA | 26:25 | R/W | Set if the inner Q-tag must be passed on to the S1 TCAM instead of the outer Q-tag. For single tagged frames, the outer tag is used. For untagged frames, the port VLAN is used. This bit influences the TPID, VID, PCP, and DEI input to the S1 key generation. | 0x0 |
| S2_UDP_PAYLOAD_ENA | 24:23 | R/W | If set, payload bytes 0, 1, 4, and 6 following the UDP header replaces the source IP address in the S2 IP4_TCP_UDP key for UDP frames. Bit 0 controls first lookup in S2 and bit 1 controls second lookup in S2. | 0x0 |
| S2_ETYPE_PAYLOAD_ENA | 22:21 | R/W | If set, payload bytes 2-7 following the EtherType replaces the source MAC address in the S2 MAC ETYPE key. Payload bytes 0-1 immediately after the EtherType are already available in the key. Bit 0 controls first lookup in S2 and bit 1 controls second lookup in S2. | 0x0 |
| S2_ENA | 20 | R/W | If S2 is enabled, each frame received on this port is processed and matched against the entries in the S2 TCAM. Each frame results in two lookups to determine both an ingress and an egress action. | 0x0 |
| S2_SNAP_DIS | 19:18 | R/W | If set, MAC_SNAP frames received on this port are treated as MAC_LLC frames when matching in S2.<br>Bit 0 controls the ingress lookup and bit 1 controls the egress lookup. | 0x0 |

*Table 293 •* **Fields in VCAP_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| S2_ARP_DIS | 17:16 | R/W | If set, MAC_ARP frames received on this port are treated as MAC_ETYPE frames when matching in S2.<br>Bit 0 controls the ingress lookup and bit 1 controls the egress lookup. | 0x0 |
| S2_IP_TCPUDP_DIS | 15:14 | R/W | If set, IP_TCPUDP frames received on this port are treated as MAC_ETYPE frames when matching in S2.<br>Bit 0 controls the ingress lookup and bit 1 controls the egress lookup. | 0x0 |
| S2_IP_OTHER_DIS | 13:12 | R/W | If set, IP_OTHER frames received on this port are treated as MAC_ETYPE frames when matching in S2.<br>Bit 0 controls the ingress lookup and bit 1 controls the egress lookup. | 0x0 |
| S2_IP6_STD_DIS | 11:10 | R/W | If set, IP6_STD frames received on this port are not matched against IP6_STD entries. If S2_IP6_TCPUDP_OTHER_DIS is set, IP6_STD frames are matched against MAC_ETYPE entries. If S2_IP6_TCPUDP_OTHER_DIS is cleared, TCP/UDP IP6_STD frames are matched against IP4_TCPUDP entries, otherwise against IP4_OTHER entries. Bit 0 controls the ingress lookup and bit 1 controls the egress lookup. | 0x0 |
| S2_IP6_TCPUDP_OTHER_DIS | 9:8 | R/W | See S2_IP6_STD_DIS for details.<br>Bit 0 controls the ingress lookup and bit 1 controls the egress lookup. | 0x0 |
| PAG_VAL | 7:0 | R/W | Default PAG value used as input to S2. The PAG value can be changed by S1 actions. | 0x00 |

### 7.4.3.5 ANA:PORT:QOS_PCP_DEI_MAP_CFG

**Parent:** ANA:PORT

**Instances:** 16

*Table 294 •* **Fields in QOS_PCP_DEI_MAP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DP_PCP_DEI_VAL | 3 | R/W | Map the frame's PCP and DEI values to a drop precedence level. DP level = QOS_PCP_DEI_MAP_CFG[index].DP_PCP_DEI_VAL, where index = 8*DEI + PCP. Only applicable to tagged frames. The use of Inner or outer tag can be selected using VLAN_CFG.VLAN_INNER_TAG_ENA. | 0x0 |
| QOS_PCP_DEI_VAL | 2:0 | R/W | Map the frame's PCP and DEI values to a QoS class. QoS class = QOS_PCP_DEI_MAP_CFG[index].QOS_PCP_DEI_VAL, where index = 8*DEI + PCP. Only applicable to tagged frames. The use of Inner or outer tag can be selected using VLAN_CFG.VLAN_INNER_TAG_ENA. | 0x0 |

### 7.4.3.6 ANA:PORT:CPU_FWD_CFG

**Parent:** ANA:PORT

**Instances:** 1

*Table 295 •* **Fields in CPU_FWD_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CPU_MLD_REDIR_ENA | 4 | R/W | If set, MLD frames are redirected to the CPU. | 0x0 |
| CPU_IGMP_REDIR_ENA | 3 | R/W | If set, IGMP frames are redirected to the CPU. | 0x0 |
| CPU_IPMC_CTRL_COPY_ENA | 2 | R/W | If set, IPv4 multicast control frames (destination IP address in the range 224.0.0.x) are copied to the CPU. | 0x0 |
| CPU_SRC_COPY_ENA | 1 | R/W | If set, all frames received on this port are copied to the CPU extraction queue given by CPUQ_CFG.CPUQ_SRC_COPY. | 0x0 |
| CPU_ALLBRIDGE_REDIR_ENA | 0 | R/W | If set, All LANs bridge management group frames (DMAC = 01-80-C2-00-00-10) are redirected to the CPU. | 0x0 |

### 7.4.3.7 ANA:PORT:CPU_FWD_BPDU_CFG

**Parent:** ANA:PORT

**Instances:** 1

***Table 296 •*** **Fields in CPU_FWD_BPDU_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BPDU_REDIR_ENA | 15:0 | R/W | If bit x is set, BPDU frame (DMAC = 01-80-C2-00-00-0x) is redirected to the CPU. | 0x0000 |

### 7.4.3.8 ANA:PORT:CPU_FWD_GARP_CFG

**Parent:** ANA:PORT

**Instances:** 1

***Table 297 •*** **Fields in CPU_FWD_GARP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GARP_REDIR_ENA | 15:0 | R/W | If bit x is set, GARP frame (DMAC = 01-80-C2-00-00-2x) is redirected to the CPU. | 0x0000 |

### 7.4.3.9 ANA:PORT:CPU_FWD_CCM_CFG

**Parent:** ANA:PORT

**Instances:** 1

***Table 298 •*** **Fields in CPU_FWD_CCM_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CCM_REDIR_ENA | 15:0 | R/W | If bit x is set, CCM/Link trace frame (DMAC = 01-80-C2-00-00-3x) is redirected to the CPU. | 0x0000 |

### 7.4.3.10 ANA:PORT:PORT_CFG

**Parent:** ANA:PORT

**Instances:** 1

***Table 299 •*** **Fields in PORT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SRC_MIRROR_ENA | 14 | R/W | If set, all frames received on this port are mirrored to the port set configured in MIRRORPORTS (ie. ingress mirroring). For egress mirroring, see EMIRRORPORTS. | 0x0 |

*Table 299 •* **Fields in PORT_CFG** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LIMIT_DROP | 13 | R/W | If set, learn frames on an ingress port, which has exceeded the maximum number of MAC table entries are discarded. Forwarding to CPU is still allowed. Note that if LEARN_ENA is cleared, then the LIMIT_DROP is ignored. | 0x0 |
| LIMIT_CPU | 12 | R/W | If set, learn frames on an ingress port, which has exceeded the maximum number of MAC table entries are copied to the CPU extraction queue specified in CPUQ_CFG.CPUQ_LRN. Note that if LEARN_ENA is cleared, then the LIMIT_CPU is ignored. | 0x0 |
| LOCKED_PORTMOVE_DROP | 11 | R/W | If set, incoming frames triggering a port move for a locked entry in the MAC table received on this port are discarded. Forwarding to CPU is still allowed. Note that if LEARN_ENA is cleared, then the LOCKED_PORTMOVE_DROP is ignored. | 0x0 |
| LOCKED_PORTMOVE_CPU | 10 | R/W | If set, incoming frames triggering a port move for a locked MAC table entry received on this port are copied to the CPU extraction queue specified in CPUQ_CFG.CPUQ_LOCKED_PORTMOVE. Note that if LEARN_ENA is cleared, then the LOCKED_PORTMOVE_CPU is ignored. | 0x0 |
| LEARNDROP | 9 | R/W | If set, incoming learn frames received on this port are discarded. Forwarding to CPU is still allowed. Note that if LEARN_ENA is cleared, then the LEARNDROP is ignored. | 0x0 |
| LEARNCPU | 8 | R/W | If set, incoming learn frames received on this port are copied to the CPU extraction queue specified in AGENCTRL.CPUQ_LRN. Note that if LEARN_ENA is cleared, then the LEARNCPU is ignored. | 0x0 |
| LEARNAUTO | 7 | R/W | If set, incoming learn frames received on this port are auto learned. Note that if LEARN_ENA is cleared, then the LEARNAUTO is ignored. | 0x1 |

*Table 299 •* **Fields in PORT_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LEARN_ENA | 6 | R/W | Enable learning for frames received on this port. If cleared, learning is skipped and any configuration settings in LEARNAUTO, LEARNCPU, LEARNDROP is ignored. | 0x1 |
| RECV_ENA | 5 | R/W | Enable reception of frames. If cleared, all incoming frames on this port are discarded by the analyzer. | 0x1 |
| PORTID_VAL | 4:0 | R/W | Logical port number for front port. If port is not a member of a LLAG, then PORTID must be set to the physical port number.<br>If port is a member of a LLAG, then PORTID must be set to the common PORTID_VAL used for all member ports of the LLAG. | 0x00 |

### 7.4.3.11 ANA:PORT:POL_CFG

**Parent:** ANA:PORT

**Instances:** 1

*Table 300 •* **Fields in POL_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| POL_CPU_REDIR_8021 | 19 | R/W | If set, frames with a DMAC = IEEE reserved addresses (BPDU, GARP, CCM, ALLBRIGDE), which are redirected to the CPU are not policed by any policers. The frames are still counted in the policer buckets. | 0x0 |
| POL_CPU_REDIR_IP | 18 | R/W | If set, IGMP and MLD frames, which are redirected to the CPU are not policed by any policers. The frames are still counted in the policers buckets. | 0x0 |
| PORT_POL_ENA | 17 | R/W | Enable port policing. Port policing on port P uses policer P. | 0x0 |
| QUEUE_POL_ENA | 16:9 | R/W | Bitmask, where bit<n> enables policing of frames classified to QoS class n on this port. Queue policing of QoS class Q on port P uses policer 32+P*8+Q. | 0x00 |

*Table 300 •* **Fields in POL_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| POL_ORDER | 8:0 | R/W | Each frame is checked against three policers: PORT(0), QOS/PORT(1) and ACL(2). In this register, a bit set will make updating of a policer be dependant on the result from another.<br><br>Bit<n+3*m> set means: Policer state <n> is checked before policer <m> is updated.<br><br>Bit0: Port policer must be open in order to update port policer with frame<br>Bit1: QoS policer must be open in order to update port policer with frame<br>Bit2: ACL policer must be open in order to update port policer with frame<br><br>Bit3: Port policer must be open in order to update QoS policer with frame<br>Bit4: QoS policer must be open in order to update QoS policer with frame<br>Bit5: ACL policer must be open in order to update QoS policer with frame<br><br>Bit6: Port policer must be open in order to update ACL policer with frame<br>Bit7: QoS policer must be open in order to update ACL policer with frame<br>Bit8: ACL policer must be open in order to update ACL policer with frame | 0x1FF |

## 7.4.4 ANA:COMMON

**Parent:** ANA

**Instances:** 1

*Table 301* • **Registers in COMMON**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| AGGR_CFG | 0x00000000 | 1 | Aggregation code generation | Page 300 |
| CPUQ_CFG | 0x00000004 | 1 | CPU extraction queue configuration | Page 301 |
| CPUQ_8021_CFG | 0x00000008 | 16<br>0x00000004 | CPU extraction queue per address of BPDU, GARP, and CCM frames. | Page 301 |
| DSCP_CFG | 0x00000048 | 64<br>0x00000004 | DSCP configuration per DSCP value. | Page 302 |
| DSCP_REWR_CFG | 0x00000148 | 16<br>0x00000004 | DSCP rewrite values per DP level and QoS class | Page 302 |
| VCAP_RNG_TYPE_CFG | 0x00000188 | 8<br>0x00000004 | VCAP range checkers | Page 303 |
| VCAP_RNG_VAL_CFG | 0x000001A8 | 8<br>0x00000004 | Range configuration per range checker | Page 303 |

### 7.4.4.1 ANA:COMMON:AGGR_CFG

**Parent:** ANA:COMMON

**Instances:** 1

*Table 302* • **Fields in AGGR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AC_RND_ENA | 6 | R/W | Use pseudo random number for aggregation code. Overrule other contributions. | 0x0 |
| AC_DMAC_ENA | 5 | R/W | Use the lower 12 bits of the destination MAC address for aggregation code. | 0x0 |
| AC_SMAC_ENA | 4 | R/W | Use the lower 12 bits of the source MAC address for aggregation code. | 0x0 |
| AC_IP6_FLOW_LBL_ENA | 3 | R/W | Use the 20-bit IPv6 flow label for aggregation code. | 0x0 |
| AC_IP6_TCPUDP_ENA | 2 | R/W | Use least significant 8 bits of both source port and destination port of IPv6 frames for aggregation code. | 0x0 |
| AC_IP4_SIPDIP_ENA | 1 | R/W | Use least significant 8 bits of both source IP address and destination IP address of IPv4 frames for aggregation code. | 0x0 |

*Table 302 •* **Fields in AGGR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AC_IP4_TCPUDP_ENA | 0 | R/W | Use least significant 8 bits of both source port and destination port of IPv4 frames for aggregation code. | 0x0 |

### 7.4.4.2 ANA:COMMON:CPUQ_CFG

**Parent:** ANA:COMMON

**Instances:** 1

*Table 303 •* **Fields in CPUQ_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CPUQ_MLD | 29:27 | R/W | CPU extraction queue used for MLD frames. | 0x0 |
| CPUQ_IGMP | 26:24 | R/W | CPU extraction queue used for IGMP frames. | 0x0 |
| CPUQ_IPMC_CTRL | 23:21 | R/W | CPU extraction queue used for IPv4 multicast control frames. | 0x0 |
| CPUQ_ALLBRIDGE | 20:18 | R/W | CPU extraction queue used for allbridge frames (DMAC = 01-80-C2-00-00-10). | 0x0 |
| CPUQ_LOCKED_PORTMOVE | 17:15 | R/W | CPU extraction queue for frames triggering a port move for a locked MAC table entry. | 0x0 |
| CPUQ_SRC_COPY | 14:12 | R/W | CPU extraction queue for frames copied due to CPU_SRC_COPY_ENA | 0x0 |
| CPUQ_MAC_COPY | 11:9 | R/W | CPU extraction queue for frames copied due to CPU_COPY return by MAC table lookup | 0x0 |
| CPUQ_LRN | 8:6 | R/W | CPU extraction queue for frames copied due to learned or moved stations. | 0x0 |
| CPUQ_MIRROR | 5:3 | R/W | CPU extraction queue for frames copied due to mirroring to the CPU. | 0x0 |
| CPUQ_SFLOW | 2:0 | R/W | CPU extraction queue for frames copied due to SFLOW sampling. | 0x0 |

### 7.4.4.3 ANA:COMMON:CPUQ_8021_CFG

**Parent:** ANA:COMMON

**Instances:** 16

*Table 304* • **Fields in CPUQ_8021_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CPUQ_BPDU_VAL | 8:6 | R/W | CPU extraction queue used for BPDU frames. | 0x0 |
| CPUQ_GARP_VAL | 5:3 | R/W | CPU extraction queue used for GARP frames. | 0x0 |
| CPUQ_CCM_VAL | 2:0 | R/W | CPU extraction queue used for CCM/Link trace frames. | 0x0 |

### 7.4.4.4 ANA:COMMON:DSCP_CFG

**Parent:** ANA:COMMON

**Instances:** 64

*Table 305* • **Fields in DSCP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DP_DSCP_VAL | 11 | R/W | Maps the frame's DSCP value to a drop precedence level. This is enabled in QOS_CFG.QOS_DSCP_ENA. | 0x0 |
| QOS_DSCP_VAL | 10:8 | R/W | Maps the frame's DSCP value to a QoS class. This is enabled in QOS_CFG.QOS_DSCP_ENA. | 0x0 |
| DSCP_TRANSLATE_VAL | 7:2 | R/W | Translated DSCP value triggered if DSCP translation is set for port (QOS_CFG[port].DSCP_TRANSLATE_ENA) | 0x00 |
| DSCP_TRUST_ENA | 1 | R/W | Must be set for a DSCP value if the DSCP value is to be used for QoS classification. | 0x0 |
| DSCP_REWR_ENA | 0 | R/W | Set if the DSCP value is selected to be rewritten. This is controlled in QOS_CFG.DSCP_REWR_CFG. | 0x0 |

### 7.4.4.5 ANA:COMMON:DSCP_REWR_CFG

**Parent:** ANA:COMMON

**Instances:** 16

*Table 306* • **Fields in DSCP_REWR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DSCP_QOS_REWR_VAL | 5:0 | R/W | Map the frame's DP level and QoS class to a DSCP value. DSCP = DSCP_REWR_CFG[8*DP level + QoS class].DSCP_QOS_REWR_VAL. This is controlled in QOS_CFG.DSCP_REWR_CFG and DSCP_CFG.DSCP_REWR_ENA. | 0x00 |

### 7.4.4.6 ANA:COMMON:VCAP_RNG_TYPE_CFG

**Parent:** ANA:COMMON

**Instances:** 8

*Table 307* • **Fields in VCAP_RNG_TYPE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VCAP_RNG_CFG | 2:0 | R/W | 0: Idle<br>1: TCP/UDP destination port is matched against range<br>2: TCP/UDP source port is matched against range<br>3: TCP/UDP source and destination ports are matched against range. Match if either source or destination port is within range.<br>4: VID is matched against range (S1: VID in frame, S2: classified VID)<br>5: DSCP value is matched against range<br>6: Reserved<br>7: Reserved | 0x0 |

### 7.4.4.7 ANA:COMMON:VCAP_RNG_VAL_CFG

**Parent:** ANA:COMMON

**Instances:** 8

*Table 308* • **Fields in VCAP_RNG_VAL_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VCAP_RNG_MIN_VAL | 31:16 | R/W | Lower value. Value is included in range. | 0x0000 |

*Table 308 •* **Fields in VCAP_RNG_VAL_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VCAP_RNG_MAX_VAL | 15:0 | R/W | Upper value. Value is included in range. | 0x0000 |

# 7.5 REW

*Table 309 •* **Register Groups in REW**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PORT | 0x00000000 | 28<br>0x00000080 | Per port configurations for Rewriter | Page 304 |
| COMMON | 0x00000E00 | 1 | Common configurations for Rewriter | Page 307 |

## 7.5.1 REW:PORT

**Parent:** REW

**Instances:** 28

*Table 310 •* **Registers in PORT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PORT_VLAN_CFG | 0x00000000 | 1 | Port VLAN configuration | Page 304 |
| TAG_CFG | 0x00000004 | 1 | Tagging configuration | Page 305 |
| PORT_CFG | 0x00000008 | 1 | Special port configuration | Page 305 |
| DSCP_CFG | 0x0000000C | 1 | DSCP updates | Page 306 |
| PCP_DEI_QOS_MAP_CFG | 0x00000010 | 16<br>0x00000004 | Mapping of DP level and QoS class to PCP and DEI values. | Page 307 |

### 7.5.1.1 REW:PORT:PORT_VLAN_CFG

**Parent:** REW:PORT

**Instances:** 1

*Table 311 •* **Fields in PORT_VLAN_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PORT_TPID | 31:16 | R/W | Tag Protocol Identifier for port. | 0x0000 |

*Table 311 •* **Fields in PORT_VLAN_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PORT_DEI | 15 | R/W | DEI value for port when TAG_CFG_TAG_QOS_TAG = 2. Otherwise, the DP level is used as the DEI value for the port. | 0x0 |
| PORT_PCP | 14:12 | R/W | PCP value for port. | 0x0 |
| PORT_VID | 11:0 | R/W | VID value for port. | 0x001 |

### 7.5.1.2 REW:PORT:TAG_CFG

**Parent:** REW:PORT

**Instances:** 1

*Table 312 •* **Fields in TAG_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TAG_CFG | 6:5 | R/W | Enable VLAN port tagging.<br>0: Port tagging disabled.<br>1: Tag all frames, except when VID=PORT_VLAN_CFG.PORT_VID or VID=0.<br>2: Tag all frames, except when VID=0.<br>3: Tag all frames. | 0x0 |
| TAG_TPID_CFG | 4:3 | R/W | Select TPID EtherType in port tag.<br>0: Use 0x8100.<br>1: Use 0x88A8.<br>2: Use custom value from PORT_VLAN_CFG.PORT_TPID.<br>3: Use PORT_VLAN_CFG.PORT_TPID, unless ingress tag was a C-tag (EtherType = 0x8100) | 0x0 |
| TAG_VID_CFG | 2 | R/W | Select VID in port tag. It can be set to either the classified VID or VID_A_VAL from the ES0 service action.<br>0: Use classified VID.<br>1: Use VID_A_VAL from ES0 action if hit, otherwise use classified VID. | 0x0 |
| TAG_QOS_CFG | 1:0 | R/W | Select PCP/DEI fields in port tag.<br>0: Use classified PCP/DEI values.<br>1: Use PCP/DEI values from ES0 action if hit, otherwise classified values.<br>2: Use PCP/DEI values from port VLAN tag in PORT_VLAN_CFG.<br>3: Use DP level and QoS class mapped to PCP/DEI values (PCP_DEI_QOS_MAP_CFG). | 0x0 |

### 7.5.1.3 REW:PORT:PORT_CFG

**Parent:** REW:PORT

**Instances:** 1

*Table 313 •* **Fields in PORT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ES0_ENA | 8 | R/W | Enable ES0 lookup. | 0x0 |
| IFH_INSERT_ENA | 7 | R/W | Insert IFH into frame (mainly for CPU ports) | 0x0 |
| IFH_INSERT_MODE | 6 | R/W | Select the position of IFH in the generated frames when IFH_INSERT_ENA is set<br>0: IFH written before DMAC.<br>1: IFH written after SMAC. | 0x0 |
| FCS_UPDATE_NONCPU_CFG | 5:4 | R/W | FCS update mode for frames not received on the CPU port.<br>0: Update FCS if frame data has changed<br>1: Never update FCS<br>2: Always update FCS | 0x0 |
| FCS_UPDATE_CPU_ENA | 3 | R/W | If set, update FCS for all frames injected by the CPU. If cleared, never update the FCS. | 0x1 |
| FLUSH_ENA | 2 | R/W | If set, all frames destined for the egress port are discarded.<br>**Note** Flushing must be disabled on ports operating in half-duplex mode. | 0x0 |
| AGE_DIS | 1 | R/W | Disable frame ageing for this egress port.<br>**Note** Frame ageing must be disabled on ports operating in half-duplex mode. | 0x0 |

### 7.5.1.4 REW:PORT:DSCP_CFG

**Parent:** REW:PORT

**Instances:** 1

*Table 314 •* **Fields in DSCP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DSCP_REWR_CFG | 1:0 | R/W | Egress DSCP rewrite.<br><br>0: No update of DSCP value in frame.<br>1: Update with DSCP value from analyzer.<br>2: Update with DSCP value from analyzer remapped through DSCP_REMAP_CFG.<br>3. Update with DSCP value from analyzer remapped based on drop precedence level through DSCP_REMAP_CFG or DSCP_REMAP_DP1_CFG. | 0x0 |

#### 7.5.1.5 REW:PORT:PCP_DEI_QOS_MAP_CFG

**Parent:** REW:PORT

**Instances:** 16

*Table 315 •* **Fields in PCP_DEI_QOS_MAP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DEI_QOS_VAL | 3 | R/W | Map the frame's DP level and QoS class to a DEI value. DEI = PCP_DEI_QOS_MAP_CFG[8*DP level + QoS class].DEI_QOS_VAL. This must be enabled in VLAN_CFG.QOS_CFG. | 0x0 |
| PCP_QOS_VAL | 2:0 | R/W | Map the frame's DP level and QoS class to a PCP value. PCP = PCP_DEI_QOS_MAP_CFG[8*DP level + QoS class].PCP_QOS_VAL. This must be enabled in VLAN_CFG.QOS_CFG. | 0x0 |

### 7.5.2 REW:COMMON

**Parent:** REW

**Instances:** 1

*Table 316 •* **Registers in COMMON**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| DSCP_REMAP_DP1_CFG | 0x00000000 | 64<br>0x00000004 | Remap table of DSCP values for frames with drop precedence set | Page 308 |
| DSCP_REMAP_CFG | 0x00000100 | 64<br>0x00000004 | Remap table of DSCP values. | Page 308 |

### 7.5.2.1 REW:COMMON:DSCP_REMAP_DP1_CFG

**Parent:** REW:COMMON

**Instances:** 64

*Table 317 •* **Fields in DSCP_REMAP_DP1_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DSCP_REMAP_DP1_VAL | 5:0 | R/W | One to one DSCP remapping table common for all ports. This table is used if DSCP_CFG.DSCP_REWR_ENA=3 and DP=1. | 0x00 |

### 7.5.2.2 REW:COMMON:DSCP_REMAP_CFG

**Parent:** REW:COMMON

**Instances:** 64

*Table 318 •* **Fields in DSCP_REMAP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DSCP_REMAP_VAL | 5:0 | R/W | One to one DSCP remapping table common for all ports. This table is used if DSCP_CFG.DSCP_REWR_ENA=2 or if DSCP_CFG.DSCP_REWR_ENA=3 and DP=0. | 0x00 |

## 7.6 VCAP_CORE

*Table 319 •* **Register Groups in VCAP_CORE**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_CORE_CFG | 0x00000000 | 1 | | Page 309 |

*Table 319 •* **Register Groups in VCAP_CORE *(continued)***

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_CORE_CACHE | 0x00000008 | 1 | | Page 313 |
| VCAP_CORE_STICKY | 0x0000020C | 1 | | Page 316 |
| VCAP_CONST | 0x00000210 | 1 | | Page 317 |
| TCAM_BIST | 0x0000022C | 1 | Build in test for TCAM | Page 319 |

## 7.6.1 VCAP_CORE:VCAP_CORE_CFG

**Parent:** VCAP_CORE

**Instances:** 1

*Table 320 •* **Registers in VCAP_CORE_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_UPDATE_CTRL | 0x00000000 | 1 | | Page 309 |
| VCAP_MV_CFG | 0x00000004 | 1 | | Page 313 |

### 7.6.1.1 VCAP_CORE:VCAP_CORE_CFG:VCAP_UPDATE_CTRL

**Parent:** VCAP_CORE:VCAP_CORE_CFG

**Instances:** 1

*Table 321 •* **Fields in VCAP_UPDATE_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UPDATE_CMD | 24:22 | R/W | Specifies the operation to be carried out when the vcap_update_shot field is set.<br><br>The COPY operations will read or write the content of the VCAP_CORE_CACHE to the TCAM/RAMs at the address specified in UPDATE_ADDR. When writing default actions UPDATE_ENTRY_DIS must be set to avoid overwriting entries at the low addresses in the TCAM.<br><br>The MOVE operations will move one or more rows up or down starting from the address in UPDATE_ADDR. The number of rows to move is specified in VCAP_MV_CFG.MV_NUM_POS. Moving a row up will decrease its address by MV_NUM_POS, moving a row down will increase its address by MV_NUM_POS. The number of rows to include in the move is specified in VCAP_MV_CFG.MV_SIZE. If a row is moved to an destination address that is less than zero, i.e. if UPDATE_ADDR - MV_NUM_POS < 0, the row will be invalidated in the TCAM and the action and counter RAM will be set to zero.<br><br>The INIT operations will set the range of rows specified by UPDATE_ADDR and VCAP_MV_CFG.MV_SIZE to the value of cache.<br><br>Note: init starts from the address specified in UPDATE_ADDR.<br>000: Copy entry and/or action from cache to TCAM/RAM<br>001: Copy entry and/or action from TCAM/RAM to cache<br>010: Move entry and/or action up (decreasing addresses)<br>011: Move entry and/or action down (increasing addresses)<br>100: Initialize all entries and/or actions with the value in the cache. | 0x0 |

*Table 321 •* **Fields in VCAP_UPDATE_CTRL** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UPDATE_ENTRY_DIS | 21 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to entries.<br>0: Entries are copied/moved/init.<br>1: Entries are not copied/moved/init. | 0x0 |
| UPDATE_ACTION_DIS | 20 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to actions.<br>0: Actions are copied/moved/init.<br>1: Actions are not copied/moved/init. | 0x0 |
| UPDATE_CNT_DIS | 19 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to cnts.<br>0: Counters are copied/moved/init.<br>1: Counters are not copied/moved/init. | 0x0 |
| UPDATE_ADDR | 18:3 | R/W | The entry/action address (row-wise) used for copy/move operations.<br><br>When accessing the default actions the offset specified in VCAP_CONST:ENTRY_CNT should be added to the default action addr, i.e.<br>Default action 0: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + 0.<br>Default action 1: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + 1.<br>Default action n: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + n. | 0x0000 |
| UPDATE_SHOT | 2 | One-shot | Initiate the operation specified in vcap_update_cmd. The bit is cleared by hw when operation has finished. | 0x0 |
| CLEAR_CACHE | 1 | One-shot | Reset all registers in VCAP_CORE_CACHE. The register is cleared by hw when the operation has finished. | 0x0 |

*Table 321 •* **Fields in VCAP_UPDATE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MV_TRAFFIC_IGN | 0 | R/W | Ignore interrupting traffic during move operation. If a lookup is performed during a move operation the move is finished from where it was interrupted. When this field is set counters are not guaranteed to be up-to-date after the move. | 0x0 |

### 7.6.1.2 VCAP_CORE:VCAP_CORE_CFG:VCAP_MV_CFG

**Parent:** VCAP_CORE:VCAP_CORE_CFG

**Instances:** 1

*Table 322 •* **Fields in VCAP_MV_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MV_NUM_POS | 31:16 | R/W | Specifies the number of positions the row must be moved up or down during a move operation.<br><br>0x0 The row is moved one position up or down.<br>0x1: The row is moved two positions up or down.<br>0xn: The row is moved n+1 positions up or down. | 0x0000 |
| MV_SIZE | 15:0 | R/W | Specifies the number of rows that must be moved up or down during a move operation.<br>This field is also used to define the range that is initialized using the init feature.<br>0x0: The row at address UPDATE_ADDR is moved up or down.<br>0x1: The row at address UPDATE_ADDR through UPDATE_ADDR+1 are moved up or down.<br>0x2: The row at address UPDATE_ADDR through UPDATE_ADDR+2 are moved up or down.<br>0xn: The row at address UPDATE_ADDR through UPDATE_ADDR+n are moved up or down. | 0x0000 |

## 7.6.2 VCAP_CORE:VCAP_CORE_CACHE

**Parent:** VCAP_CORE

**Instances:** 1

*Table 323* • **Registers in VCAP_CORE_CACHE**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_ENTRY_DAT | 0x00000000 | 32<br>0x00000004 | | Page 314 |
| VCAP_MASK_DAT | 0x00000080 | 32<br>0x00000004 | | Page 314 |
| VCAP_ACTION_DAT | 0x00000100 | 32<br>0x00000004 | | Page 315 |
| VCAP_CNT_DAT | 0x00000180 | 32<br>0x00000004 | | Page 315 |
| VCAP_TG_DAT | 0x00000200 | 1 | | Page 316 |

### 7.6.2.1 VCAP_CORE:VCAP_CORE_CACHE:VCAP_ENTRY_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 324* • **Fields in VCAP_ENTRY_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_DAT | 31:0 | R/W | The cache register that holds a single TCAM entry data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.ENTRY_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.6.2.2 VCAP_CORE:VCAP_CORE_CACHE:VCAP_MASK_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 325 •* **Fields in VCAP_MASK_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MASK_DAT | 31:0 | R/W | The cache register that holds a single TCAM entry mask row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the mask cache is specified in VCAP_CONST.ENTRY_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.6.2.3 VCAP_CORE:VCAP_CORE_CACHE:VCAP_ACTION_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 326 •* **Fields in VCAP_ACTION_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_DAT | 31:0 | R/W | The cache register that holds a single action ram data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.ACTION_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.6.2.4 VCAP_CORE:VCAP_CORE_CACHE:VCAP_CNT_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 327 •* **Fields in VCAP_CNT_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT_DAT | 31:0 | R/W | The cache register that holds a single counter ram data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.CNT_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.6.2.5 VCAP_CORE:VCAP_CORE_CACHE:VCAP_TG_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 1

*Table 328 •* **Fields in VCAP_TG_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TG_DAT | 31:0 | R/W | This cache register holds the TypeGroup id for each subword in the TCAM. If the VCAP supports multiple subwords, i.e. VCAP_CONST.ENTRY_SWCNT > 1, the TypeGroup ids are place back to back with subword 0 at the LSBs. | 0x00000000 |

## 7.6.3 VCAP_CORE:VCAP_CORE_STICKY

**Parent:** VCAP_CORE

**Instances:** 1

*Table 329 •* **Registers in VCAP_CORE_STICKY**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_STICKY | 0x00000000 | 1 | | Page 316 |

### 7.6.3.1 VCAP_CORE:VCAP_CORE_STICKY:VCAP_STICKY

**Parent:** VCAP_CORE:VCAP_CORE_STICKY

**Instances:** 1

*Table 330 •* **Fields in VCAP_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VCAP_ROW_DELETED_STICKY | 0 | Sticky | A move operation has resulted in one or more rows have been deleted. | 0x0 |

## 7.6.4 VCAP_CORE:VCAP_CONST

**Parent:** VCAP_CORE

**Instances:** 1

*Table 331 •* **Registers in VCAP_CONST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ENTRY_WIDTH | 0x00000000 | 1 | | Page 317 |
| ENTRY_CNT | 0x00000004 | 1 | | Page 317 |
| ENTRY_SWCNT | 0x00000008 | 1 | | Page 318 |
| ENTRY_TG_WIDTH | 0x0000000C | 1 | | Page 318 |
| ACTION_DEF_CNT | 0x00000010 | 1 | | Page 318 |
| ACTION_WIDTH | 0x00000014 | 1 | | Page 318 |
| CNT_WIDTH | 0x00000018 | 1 | | Page 318 |

### 7.6.4.1 VCAP_CORE:VCAP_CONST:ENTRY_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 332 •* **Fields in ENTRY_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_WIDTH | 9:0 | R/O | The width of a TCAM entry including TypeGroup ids. | 0x000 |

### 7.6.4.2 VCAP_CORE:VCAP_CONST:ENTRY_CNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 333 •* **Fields in ENTRY_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_CNT | 9:0 | R/O | The number of entries in the TCAM. | 0x000 |

### 7.6.4.3 VCAP_CORE:VCAP_CONST:ENTRY_SWCNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 334* • **Fields in ENTRY_SWCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_SWCNT | 5:0 | R/O | The number of supported subwords in the TCAM. | 0x00 |

### 7.6.4.4 VCAP_CORE:VCAP_CONST:ENTRY_TG_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 335* • **Fields in ENTRY_TG_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_TG_WIDTH | 5:0 | R/O | The width of a single TypeGroup id. | 0x00 |

### 7.6.4.5 VCAP_CORE:VCAP_CONST:ACTION_DEF_CNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 336* • **Fields in ACTION_DEF_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_DEF_CNT | 9:0 | R/O | The number of default actions. | 0x000 |

### 7.6.4.6 VCAP_CORE:VCAP_CONST:ACTION_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 337* • **Fields in ACTION_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_WIDTH | 9:0 | R/O | The width of the action RAM. | 0x000 |

### 7.6.4.7 VCAP_CORE:VCAP_CONST:CNT_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 338 •* **Fields in CNT_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT_WIDTH | 9:0 | R/O | The width of the counter RAM. | 0x000 |

## 7.6.5 VCAP_CORE:TCAM_BIST

**Parent:** VCAP_CORE

**Instances:** 1

*Table 339 •* **Registers in TCAM_BIST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| TCAM_CTRL | 0x00000000 | 1 | Control of the TCAM | Page 319 |
| TCAM_STAT | 0x0000000C | 1 | Status for the TCAM | Page 319 |

### 7.6.5.1 VCAP_CORE:TCAM_BIST:TCAM_CTRL

**Parent:** VCAP_CORE:TCAM_BIST

**Instances:** 1

*Table 340 •* **Fields in TCAM_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TCAM_INIT | 0 | One-shot | Set this field to start manual initialization of the TCAM. This field is cleared once initialization is complete.<br>The TCAM has random contents after reset and must be initialized prior to usage. | 0x0 |

### 7.6.5.2 VCAP_CORE:TCAM_BIST:TCAM_STAT

**Parent:** VCAP_CORE:TCAM_BIST

**Instances:** 1

*Table 341 •* **Fields in TCAM_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TCAM_RDY | 0 | R/O | Indicates the current operational state of the TCAM.<br>'0': Busy with initialization.<br>'1': Ready to be used. | 0x0 |

# 7.7 VCAP_CORE

*Table 342* • **Register Groups in VCAP_CORE**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_CORE_CFG | 0x00000000 | 1 | | Page 309 |
| VCAP_CORE_CACHE | 0x00000008 | 1 | | Page 313 |
| VCAP_CORE_STICKY | 0x0000020C | 1 | | Page 316 |
| VCAP_CONST | 0x00000210 | 1 | | Page 317 |
| TCAM_BIST | 0x0000022C | 1 | Build in test for TCAM | Page 319 |

## 7.7.1 VCAP_CORE:VCAP_CORE_CFG

**Parent:** VCAP_CORE

**Instances:** 1

*Table 343* • **Registers in VCAP_CORE_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_UPDATE_CTRL | 0x00000000 | 1 | | Page 309 |
| VCAP_MV_CFG | 0x00000004 | 1 | | Page 313 |

### 7.7.1.1 VCAP_CORE:VCAP_CORE_CFG:VCAP_UPDATE_CTRL

**Parent:** VCAP_CORE:VCAP_CORE_CFG

**Instances:** 1

*Table 344 •* **Fields in VCAP_UPDATE_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UPDATE_CMD | 24:22 | R/W | Specifies the operation to be carried out when the vcap_update_shot field is set.<br><br>The COPY operations will read or write the content of the VCAP_CORE_CACHE to the TCAM/RAMs at the address specified in UPDATE_ADDR. When writing default actions UPDATE_ENTRY_DIS must be set to avoid overwriting entries at the low addresses in the TCAM.<br><br>The MOVE operations will move one or more rows up or down starting from the address in UPDATE_ADDR. The number of rows to move is specified in VCAP_MV_CFG.MV_NUM_POS. Moving a row up will decrease its address by MV_NUM_POS, moving a row down will increase its address by MV_NUM_POS. The number of rows to include in the move is specified in VCAP_MV_CFG.MV_SIZE. If a row is moved to an destination address that is less than zero, i.e. if UPDATE_ADDR - MV_NUM_POS < 0, the row will be invalidated in the TCAM and the action and counter RAM will be set to zero.<br><br>The INIT operations will set the range of rows specified by UPDATE_ADDR and VCAP_MV_CFG.MV_SIZE to the value of cache.<br><br>Note: init starts from the address specified in UPDATE_ADDR.<br>000: Copy entry and/or action from cache to TCAM/RAM<br>001: Copy entry and/or action from TCAM/RAM to cache<br>010: Move entry and/or action up (decreasing addresses)<br>011: Move entry and/or action down (increasing addresses)<br>100: Initialize all entries and/or actions with the value in the cache. | 0x0 |

*Table 344 •* **Fields in VCAP_UPDATE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UPDATE_ENTRY_DIS | 21 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to entries.<br>0: Entries are copied/moved/init.<br>1: Entries are not copied/moved/init. | 0x0 |
| UPDATE_ACTION_DIS | 20 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to actions.<br>0: Actions are copied/moved/init.<br>1: Actions are not copied/moved/init. | 0x0 |
| UPDATE_CNT_DIS | 19 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to cnts.<br>0: Counters are copied/moved/init.<br>1: Counters are not copied/moved/init. | 0x0 |
| UPDATE_ADDR | 18:3 | R/W | The entry/action address (row-wise) used for copy/move operations.<br><br>When accessing the default actions the offset specified in VCAP_CONST:ENTRY_CNT should be added to the default action addr, i.e.<br>Default action 0: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + 0.<br>Default action 1: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + 1.<br>Default action n: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + n. | 0x0000 |
| UPDATE_SHOT | 2 | One-shot | Initiate the operation specified in vcap_update_cmd. The bit is cleared by hw when operation has finished. | 0x0 |
| CLEAR_CACHE | 1 | One-shot | Reset all registers in VCAP_CORE_CACHE. The register is cleared by hw when the operation has finished. | 0x0 |

*Table 344 •* **Fields in VCAP_UPDATE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MV_TRAFFIC_IGN | 0 | R/W | Ignore interrupting traffic during move operation. If a lookup is performed during a move operation the move is finished from where it was interrupted. When this field is set counters are not guaranteed to be up-to-date after the move. | 0x0 |

### 7.7.1.2 VCAP_CORE:VCAP_CORE_CFG:VCAP_MV_CFG

**Parent:** VCAP_CORE:VCAP_CORE_CFG

**Instances:** 1

*Table 345 •* **Fields in VCAP_MV_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MV_NUM_POS | 31:16 | R/W | Specifies the number of positions the row must be moved up or down during a move operation.<br><br>0x0 The row is moved one position up or down.<br>0x1: The row is moved two positions up or down.<br>0xn: The row is moved n+1 positions up or down. | 0x0000 |
| MV_SIZE | 15:0 | R/W | Specifies the number of rows that must be moved up or down during a move operation.<br>This field is also used to define the range that is initialized using the init feature.<br>0x0: The row at address UPDATE_ADDR is moved up or down.<br>0x1: The row at address UPDATE_ADDR through UPDATE_ADDR+1 are moved up or down.<br>0x2: The row at address UPDATE_ADDR through UPDATE_ADDR+2 are moved up or down.<br>0xn: The row at address UPDATE_ADDR through UPDATE_ADDR+n are moved up or down. | 0x0000 |

## 7.7.2 VCAP_CORE:VCAP_CORE_CACHE

**Parent:** VCAP_CORE

**Instances:** 1

*Table 346* • **Registers in VCAP_CORE_CACHE**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_ENTRY_DAT | 0x00000000 | 32<br>0x00000004 | | Page 314 |
| VCAP_MASK_DAT | 0x00000080 | 32<br>0x00000004 | | Page 314 |
| VCAP_ACTION_DAT | 0x00000100 | 32<br>0x00000004 | | Page 315 |
| VCAP_CNT_DAT | 0x00000180 | 32<br>0x00000004 | | Page 315 |
| VCAP_TG_DAT | 0x00000200 | 1 | | Page 316 |

### 7.7.2.1 VCAP_CORE:VCAP_CORE_CACHE:VCAP_ENTRY_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 347* • **Fields in VCAP_ENTRY_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_DAT | 31:0 | R/W | The cache register that holds a single TCAM entry data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.ENTRY_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.7.2.2 VCAP_CORE:VCAP_CORE_CACHE:VCAP_MASK_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 348* • **Fields in VCAP_MASK_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MASK_DAT | 31:0 | R/W | The cache register that holds a single TCAM entry mask row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the mask cache is specified in VCAP_CONST.ENTRY_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.7.2.3 VCAP_CORE:VCAP_CORE_CACHE:VCAP_ACTION_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 349* • **Fields in VCAP_ACTION_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_DAT | 31:0 | R/W | The cache register that holds a single action ram data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.ACTION_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.7.2.4 VCAP_CORE:VCAP_CORE_CACHE:VCAP_CNT_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 350 •* **Fields in VCAP_CNT_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT_DAT | 31:0 | R/W | The cache register that holds a single counter ram data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.CNT_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.7.2.5 VCAP_CORE:VCAP_CORE_CACHE:VCAP_TG_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 1

*Table 351 •* **Fields in VCAP_TG_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TG_DAT | 31:0 | R/W | This cache register holds the TypeGroup id for each subword in the TCAM. If the VCAP supports multiple subwords, i.e. VCAP_CONST.ENTRY_SWCNT > 1, the TypeGroup ids are place back to back with subword 0 at the LSBs. | 0x00000000 |

## 7.7.3 VCAP_CORE:VCAP_CORE_STICKY

**Parent:** VCAP_CORE

**Instances:** 1

*Table 352 •* **Registers in VCAP_CORE_STICKY**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_STICKY | 0x00000000 | 1 | | Page 316 |

### 7.7.3.1 VCAP_CORE:VCAP_CORE_STICKY:VCAP_STICKY

**Parent:** VCAP_CORE:VCAP_CORE_STICKY

**Instances:** 1

*Table 353 •* **Fields in VCAP_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VCAP_ROW_DELETED_STICKY | 0 | Sticky | A move operation has resulted in one or more rows have been deleted. | 0x0 |

## 7.7.4 VCAP_CORE:VCAP_CONST

**Parent:** VCAP_CORE

**Instances:** 1

*Table 354 •* **Registers in VCAP_CONST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ENTRY_WIDTH | 0x00000000 | 1 | | Page 317 |
| ENTRY_CNT | 0x00000004 | 1 | | Page 317 |
| ENTRY_SWCNT | 0x00000008 | 1 | | Page 318 |
| ENTRY_TG_WIDTH | 0x0000000C | 1 | | Page 318 |
| ACTION_DEF_CNT | 0x00000010 | 1 | | Page 318 |
| ACTION_WIDTH | 0x00000014 | 1 | | Page 318 |
| CNT_WIDTH | 0x00000018 | 1 | | Page 318 |

### 7.7.4.1 VCAP_CORE:VCAP_CONST:ENTRY_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 355 •* **Fields in ENTRY_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_WIDTH | 9:0 | R/O | The width of a TCAM entry including TypeGroup ids. | 0x000 |

### 7.7.4.2 VCAP_CORE:VCAP_CONST:ENTRY_CNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 356 •* **Fields in ENTRY_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_CNT | 9:0 | R/O | The number of entries in the TCAM. | 0x000 |

### 7.7.4.3 VCAP_CORE:VCAP_CONST:ENTRY_SWCNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 357* • **Fields in ENTRY_SWCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_SWCNT | 5:0 | R/O | The number of supported subwords in the TCAM. | 0x00 |

### 7.7.4.4 VCAP_CORE:VCAP_CONST:ENTRY_TG_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 358* • **Fields in ENTRY_TG_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_TG_WIDTH | 5:0 | R/O | The width of a single TypeGroup id. | 0x00 |

### 7.7.4.5 VCAP_CORE:VCAP_CONST:ACTION_DEF_CNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 359* • **Fields in ACTION_DEF_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_DEF_CNT | 9:0 | R/O | The number of default actions. | 0x000 |

### 7.7.4.6 VCAP_CORE:VCAP_CONST:ACTION_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 360* • **Fields in ACTION_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_WIDTH | 9:0 | R/O | The width of the action RAM. | 0x000 |

### 7.7.4.7 VCAP_CORE:VCAP_CONST:CNT_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 361* • **Fields in CNT_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT_WIDTH | 9:0 | R/O | The width of the counter RAM. | 0x000 |

## 7.7.5 VCAP_CORE:TCAM_BIST

**Parent:** VCAP_CORE

**Instances:** 1

*Table 362* • **Registers in TCAM_BIST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| TCAM_CTRL | 0x00000000 | 1 | Control of the TCAM | Page 319 |
| TCAM_STAT | 0x0000000C | 1 | Status for the TCAM | Page 319 |

### 7.7.5.1 VCAP_CORE:TCAM_BIST:TCAM_CTRL

**Parent:** VCAP_CORE:TCAM_BIST

**Instances:** 1

*Table 363* • **Fields in TCAM_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TCAM_INIT | 0 | One-shot | Set this field to start manual initialization of the TCAM. This field is cleared once initialization is complete.<br>The TCAM has random contents after reset and must be initialized prior to usage. | 0x0 |

### 7.7.5.2 VCAP_CORE:TCAM_BIST:TCAM_STAT

**Parent:** VCAP_CORE:TCAM_BIST

**Instances:** 1

*Table 364* • **Fields in TCAM_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TCAM_RDY | 0 | R/O | Indicates the current operational state of the TCAM.<br>'0': Busy with initialization.<br>'1': Ready to be used. | 0x0 |

# 7.8 VCAP_CORE

*Table 365* • **Register Groups in VCAP_CORE**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_CORE_CFG | 0x00000000 | 1 | | Page 309 |
| VCAP_CORE_CACHE | 0x00000008 | 1 | | Page 313 |
| VCAP_CORE_STICKY | 0x0000020C | 1 | | Page 316 |
| VCAP_CONST | 0x00000210 | 1 | | Page 317 |
| TCAM_BIST | 0x0000022C | 1 | Build in test for TCAM | Page 319 |

## 7.8.1 VCAP_CORE:VCAP_CORE_CFG

**Parent:** VCAP_CORE

**Instances:** 1

*Table 366* • **Registers in VCAP_CORE_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_UPDATE_CTRL | 0x00000000 | 1 | | Page 309 |
| VCAP_MV_CFG | 0x00000004 | 1 | | Page 313 |

### 7.8.1.1 VCAP_CORE:VCAP_CORE_CFG:VCAP_UPDATE_CTRL

**Parent:** VCAP_CORE:VCAP_CORE_CFG

**Instances:** 1

*Table 367 •* **Fields in VCAP_UPDATE_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UPDATE_CMD | 24:22 | R/W | Specifies the operation to be carried out when the vcap_update_shot field is set.<br><br>The COPY operations will read or write the content of the VCAP_CORE_CACHE to the TCAM/RAMs at the address specified in UPDATE_ADDR. When writing default actions UPDATE_ENTRY_DIS must be set to avoid overwriting entries at the low addresses in the TCAM.<br><br>The MOVE operations will move one or more rows up or down starting from the address in UPDATE_ADDR. The number of rows to move is specified in VCAP_MV_CFG.MV_NUM_POS. Moving a row up will decrease its address by MV_NUM_POS, moving a row down will increase its address by MV_NUM_POS. The number of rows to include in the move is specified in VCAP_MV_CFG.MV_SIZE. If a row is moved to an destination address that is less than zero, i.e. if UPDATE_ADDR - MV_NUM_POS < 0, the row will be invalidated in the TCAM and the action and counter RAM will be set to zero.<br><br>The INIT operations will set the range of rows specified by UPDATE_ADDR and VCAP_MV_CFG.MV_SIZE to the value of cache.<br><br>Note: init starts from the address specified in UPDATE_ADDR.<br>000: Copy entry and/or action from cache to TCAM/RAM<br>001: Copy entry and/or action from TCAM/RAM to cache<br>010: Move entry and/or action up (decreasing addresses)<br>011: Move entry and/or action down (increasing addresses)<br>100: Initialize all entries and/or actions with the value in the cache. | 0x0 |

*Table 367 •* **Fields in VCAP_UPDATE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UPDATE_ENTRY_DIS | 21 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to entries.<br>0: Entries are copied/moved/init.<br>1: Entries are not copied/moved/init. | 0x0 |
| UPDATE_ACTION_DIS | 20 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to actions.<br>0: Actions are copied/moved/init.<br>1: Actions are not copied/moved/init. | 0x0 |
| UPDATE_CNT_DIS | 19 | R/W | Specifies whether the operation specified in vcap_update_cmd is applied to cnts.<br>0: Counters are copied/moved/init.<br>1: Counters are not copied/moved/init. | 0x0 |
| UPDATE_ADDR | 18:3 | R/W | The entry/action address (row-wise) used for copy/move operations.<br><br>When accessing the default actions the offset specified in VCAP_CONST:ENTRY_CNT should be added to the default action addr, i.e.<br>Default action 0: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + 0.<br>Default action 1: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + 1.<br>Default action n: set UPDATE_ADDR to VCAP_CONST:ENTRY_CNT + n. | 0x0000 |
| UPDATE_SHOT | 2 | One-shot | Initiate the operation specified in vcap_update_cmd. The bit is cleared by hw when operation has finished. | 0x0 |
| CLEAR_CACHE | 1 | One-shot | Reset all registers in VCAP_CORE_CACHE. The register is cleared by hw when the operation has finished. | 0x0 |

*Table 367 •* **Fields in VCAP_UPDATE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MV_TRAFFIC_IGN | 0 | R/W | Ignore interrupting traffic during move operation. If a lookup is performed during a move operation the move is finished from where it was interrupted. When this field is set counters are not guaranteed to be up-to-date after the move. | 0x0 |

### 7.8.1.2 VCAP_CORE:VCAP_CORE_CFG:VCAP_MV_CFG

**Parent:** VCAP_CORE:VCAP_CORE_CFG

**Instances:** 1

*Table 368 •* **Fields in VCAP_MV_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MV_NUM_POS | 31:16 | R/W | Specifies the number of positions the row must be moved up or down during a move operation.<br><br>0x0 The row is moved one position up or down.<br>0x1: The row is moved two positions up or down.<br>0xn: The row is moved n+1 positions up or down. | 0x0000 |
| MV_SIZE | 15:0 | R/W | Specifies the number of rows that must be moved up or down during a move operation.<br>This field is also used to define the range that is initialized using the init feature.<br>0x0: The row at address UPDATE_ADDR is moved up or down.<br>0x1: The row at address UPDATE_ADDR through UPDATE_ADDR+1 are moved up or down.<br>0x2: The row at address UPDATE_ADDR through UPDATE_ADDR+2 are moved up or down.<br>0xn: The row at address UPDATE_ADDR through UPDATE_ADDR+n are moved up or down. | 0x0000 |

## 7.8.2 VCAP_CORE:VCAP_CORE_CACHE

**Parent:** VCAP_CORE

**Instances:** 1

*Table 369* • **Registers in VCAP_CORE_CACHE**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_ENTRY_DAT | 0x00000000 | 32<br>0x00000004 | | Page 314 |
| VCAP_MASK_DAT | 0x00000080 | 32<br>0x00000004 | | Page 314 |
| VCAP_ACTION_DAT | 0x00000100 | 32<br>0x00000004 | | Page 315 |
| VCAP_CNT_DAT | 0x00000180 | 32<br>0x00000004 | | Page 315 |
| VCAP_TG_DAT | 0x00000200 | 1 | | Page 316 |

### 7.8.2.1 VCAP_CORE:VCAP_CORE_CACHE:VCAP_ENTRY_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 370* • **Fields in VCAP_ENTRY_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_DAT | 31:0 | R/W | The cache register that holds a single TCAM entry data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.ENTRY_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.8.2.2 VCAP_CORE:VCAP_CORE_CACHE:VCAP_MASK_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 371 •* **Fields in VCAP_MASK_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MASK_DAT | 31:0 | R/W | The cache register that holds a single TCAM entry mask row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the mask cache is specified in VCAP_CONST.ENTRY_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.8.2.3 VCAP_CORE:VCAP_CORE_CACHE:VCAP_ACTION_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 372 •* **Fields in VCAP_ACTION_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_DAT | 31:0 | R/W | The cache register that holds a single action ram data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.ACTION_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.8.2.4 VCAP_CORE:VCAP_CORE_CACHE:VCAP_CNT_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 32

*Table 373 •* **Fields in VCAP_CNT_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT_DAT | 31:0 | R/W | The cache register that holds a single counter ram data row. The register is replicated 32 times where replication index 0 is the 32 LSBs of the cache.<br><br>Note that the physical width of the entry cache is specified in VCAP_CONST.CNT_WIDTH and that there are only instantiated flops in the CSR target for this width. | 0x00000000 |

### 7.8.2.5 VCAP_CORE:VCAP_CORE_CACHE:VCAP_TG_DAT

**Parent:** VCAP_CORE:VCAP_CORE_CACHE

**Instances:** 1

*Table 374 •* **Fields in VCAP_TG_DAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TG_DAT | 31:0 | R/W | This cache register holds the TypeGroup id for each subword in the TCAM. If the VCAP supports multiple subwords, i.e. VCAP_CONST.ENTRY_SWCNT > 1, the TypeGroup ids are place back to back with subword 0 at the LSBs. | 0x00000000 |

## 7.8.3 VCAP_CORE:VCAP_CORE_STICKY

**Parent:** VCAP_CORE

**Instances:** 1

*Table 375 •* **Registers in VCAP_CORE_STICKY**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VCAP_STICKY | 0x00000000 | 1 | | Page 316 |

### 7.8.3.1 VCAP_CORE:VCAP_CORE_STICKY:VCAP_STICKY

**Parent:** VCAP_CORE:VCAP_CORE_STICKY

**Instances:** 1

*Table 376* • **Fields in VCAP_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VCAP_ROW_DELETED_STICKY | 0 | Sticky | A move operation has resulted in one or more rows have been deleted. | 0x0 |

## 7.8.4 VCAP_CORE:VCAP_CONST

**Parent:** VCAP_CORE

**Instances:** 1

*Table 377* • **Registers in VCAP_CONST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| ENTRY_WIDTH | 0x00000000 | 1 | | Page 317 |
| ENTRY_CNT | 0x00000004 | 1 | | Page 317 |
| ENTRY_SWCNT | 0x00000008 | 1 | | Page 318 |
| ENTRY_TG_WIDTH | 0x0000000C | 1 | | Page 318 |
| ACTION_DEF_CNT | 0x00000010 | 1 | | Page 318 |
| ACTION_WIDTH | 0x00000014 | 1 | | Page 318 |
| CNT_WIDTH | 0x00000018 | 1 | | Page 318 |

### 7.8.4.1 VCAP_CORE:VCAP_CONST:ENTRY_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 378* • **Fields in ENTRY_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_WIDTH | 9:0 | R/O | The width of a TCAM entry including TypeGroup ids. | 0x000 |

### 7.8.4.2 VCAP_CORE:VCAP_CONST:ENTRY_CNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 379* • **Fields in ENTRY_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_CNT | 9:0 | R/O | The number of entries in the TCAM. | 0x000 |

### 7.8.4.3 VCAP_CORE:VCAP_CONST:ENTRY_SWCNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 380* • **Fields in ENTRY_SWCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_SWCNT | 5:0 | R/O | The number of supported subwords in the TCAM. | 0x00 |

### 7.8.4.4 VCAP_CORE:VCAP_CONST:ENTRY_TG_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 381* • **Fields in ENTRY_TG_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENTRY_TG_WIDTH | 5:0 | R/O | The width of a single TypeGroup id. | 0x00 |

### 7.8.4.5 VCAP_CORE:VCAP_CONST:ACTION_DEF_CNT

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 382* • **Fields in ACTION_DEF_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_DEF_CNT | 9:0 | R/O | The number of default actions. | 0x000 |

### 7.8.4.6 VCAP_CORE:VCAP_CONST:ACTION_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 383* • **Fields in ACTION_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTION_WIDTH | 9:0 | R/O | The width of the action RAM. | 0x000 |

### 7.8.4.7 VCAP_CORE:VCAP_CONST:CNT_WIDTH

**Parent:** VCAP_CORE:VCAP_CONST

**Instances:** 1

*Table 384 •* **Fields in CNT_WIDTH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CNT_WIDTH | 9:0 | R/O | The width of the counter RAM. | 0x000 |

## 7.8.5 VCAP_CORE:TCAM_BIST

**Parent:** VCAP_CORE

**Instances:** 1

*Table 385 •* **Registers in TCAM_BIST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| TCAM_CTRL | 0x00000000 | 1 | Control of the TCAM | Page 319 |
| TCAM_STAT | 0x0000000C | 1 | Status for the TCAM | Page 319 |

### 7.8.5.1 VCAP_CORE:TCAM_BIST:TCAM_CTRL

**Parent:** VCAP_CORE:TCAM_BIST

**Instances:** 1

*Table 386 •* **Fields in TCAM_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TCAM_INIT | 0 | One-shot | Set this field to start manual initialization of the TCAM. This field is cleared once initialization is complete.<br>The TCAM has random contents after reset and must be initialized prior to usage. | 0x0 |

### 7.8.5.2 VCAP_CORE:TCAM_BIST:TCAM_STAT

**Parent:** VCAP_CORE:TCAM_BIST

**Instances:** 1

*Table 387 •* **Fields in TCAM_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TCAM_RDY | 0 | R/O | Indicates the current operational state of the TCAM.<br>'0': Busy with initialization.<br>'1': Ready to be used. | 0x0 |

# 7.9 DEVCPU_GCB

*Table 388* • **Register Groups in DEVCPU_GCB**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CHIP_REGS | 0x00000000 | 1 | | Page 342 |
| SW_REGS | 0x00000014 | 1 | Registers for software/software interaction | Page 344 |
| VCORE_ACCESS | 0x00000054 | 1 | | Page 348 |
| GPIO | 0x00000068 | 1 | | Page 351 |
| DEVCPU_RST_REGS | 0x00000090 | 1 | | Page 355 |
| MIIM | 0x000000A0 | 2<br>0x00000024 | | Page 356 |
| MIIM_READ_SCAN | 0x000000E8 | 1 | | Page 361 |
| RAM_STAT | 0x00000114 | 1 | | Page 362 |
| MISC | 0x00000118 | 1 | Miscellaneous Registers | Page 362 |
| SIO_CTRL | 0x00000130 | 1 | Serial IO control configuration | Page 365 |
| FAN_CFG | 0x000001F0 | 1 | Configuration register for the fan controller | Page 370 |
| FAN_STAT | 0x000001F4 | 1 | Fan controller statistics | Page 371 |
| PTP_CFG | 0x000001F8 | 1 | Configuration registers for PTP | Page 371 |
| PTP_STAT | 0x00000218 | 1 | Status registers for PTP | Page 374 |
| PTP_TIMERS | 0x00000224 | 1 | Latched values of time of day timer for PTP measurements | Page 375 |
| MEMITGR | 0x00000234 | 1 | Memory integrity monitor | Page 377 |

## 7.9.1 DEVCPU_GCB:CHIP_REGS

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 389* • **Registers in CHIP_REGS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| GENERAL_PURPOSE | 0x00000000 | 1 | general purpose register | Page 343 |
| SI | 0x00000004 | 1 | SI registers | Page 343 |
| CHIP_ID | 0x00000008 | 1 | Chip Id | Page 343 |

### 7.9.1.1 DEVCPU_GCB:CHIP_REGS:GENERAL_PURPOSE

**Parent:** DEVCPU_GCB:CHIP_REGS

**Instances:** 1

*Table 390* • **Fields in GENERAL_PURPOSE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GENERAL_PURPOSE_REG | 31:0 | R/W | This is a general-purpose register that can be used for testing. The value in this register has no functionality other than general purpose storage. | 0x00000000 |

### 7.9.1.2 DEVCPU_GCB:CHIP_REGS:SI

**Parent:** DEVCPU_GCB:CHIP_REGS

**Instances:** 1

Configuration of serial interface data format. This register modifies how the SI receives and transmits data, when configuring this register first write 0 (to get to a known state), then configure the desired values.

*Table 391* • **Fields in SI**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SI_LSB | 5 | R/W | Setup SI to use MSB or LSB first. See datasheet for more information.<br>0: SI expect/transmit MSB first<br>1: SI expect/transmit LSB first | 0x0 |
| SI_ENDIAN | 4 | R/W | Setup SI to use either big or little endian data format. See datasheet for more information.<br>0: SI uses little endian notation<br>1: SI uses big endian notation | 0x1 |
| SI_WAIT_STATES | 3:0 | R/W | Configure the number of padding bytes that the SI must insert before transmitting read-data during reading from the device.<br>0 : don't insert any padding<br>1 : Insert 1 byte of padding<br>...<br>15: Insert 15 bytes of padding | 0x0 |

### 7.9.1.3 DEVCPU_GCB:CHIP_REGS:CHIP_ID

**Parent:** DEVCPU_GCB:CHIP_REGS

**Instances:** 1

*Table 392 •* **Fields in CHIP_ID**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| REV_ID | 31:28 | R/O | Revision ID. | 0x3 |
| PART_ID | 27:12 | R/O | Part ID.<br>VSC7424-02<br>VSC7425-02<br>VSC7426-02<br>VSC7427-02 | <br>0x7424<br>0x7425<br>0x7426<br>0x7427 |
| MFG_ID | 11:1 | R/O | Manufacturer's ID. | 0x074 |
| ONE | 0 | R/O | Returns '1' | 0x1 |

## 7.9.2 DEVCPU_GCB:SW_REGS

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 393 •* **Registers in SW_REGS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SEMA_INTR_ENA | 0x00000000 | 1 | Semaphore SW interrupt enable | Page 344 |
| SEMA_INTR_ENA_CLR | 0x00000004 | 1 | Clear of semaphore SW interrupt enables | Page 345 |
| SEMA_INTR_ENA_SET | 0x00000008 | 1 | Masking of semaphore | Page 345 |
| SEMA | 0x0000000C | 8<br>0x00000004 | Semaphore register | Page 346 |
| SEMA_FREE | 0x0000002C | 1 | Semaphore status | Page 346 |
| SW_INTR | 0x00000030 | 1 | Manually assert software interrupt | Page 346 |
| MAILBOX | 0x00000034 | 1 | Mailbox register | Page 347 |
| MAILBOX_CLR | 0x00000038 | 1 | Mailbox register atomic clear | Page 347 |
| MAILBOX_SET | 0x0000003C | 1 | Mailbox register atomic set | Page 347 |

### 7.9.2.1 DEVCPU_GCB:SW_REGS:SEMA_INTR_ENA

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 394 •* **Fields in SEMA_INTR_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SEMA_INTR_IDENT | 15:8 | R/O | This is a bitwise AND of SEMA_FREE and SEMA_INTR_ENA providing an fast access to the cause of an interrupt, given the current mask. | 0x00 |
| SEMA_INTR_ENA | 7:0 | R/W | Set bits in this register to enable interrupt when the corresponding semaphore is free. In a multi-threaded environment, or with more than one active processor the CPU_SEMA_ENA_SET and CPU_SEMA_ENA_CLR registers can be used for atomic modifications of this register.<br>If interrupt is enabled for a particular semaphore, then software interrupt will be asserted for as long as the semaphore is free (and interrupt is enabled for that semaphore). The lower half of the available semaphores are connected to software Interrupt 0 (SW0), the upper half is connected to software interrupt 1 (SW1). | 0x00 |

### 7.9.2.2 DEVCPU_GCB:SW_REGS:SEMA_INTR_ENA_CLR

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 395 •* **Fields in SEMA_INTR_ENA_CLR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SEMA_INTR_ENA_CLR | 7:0 | One-shot | Set to clear corresponding interrupt enable in SEMA_INTR_ENA. | 0x00 |

### 7.9.2.3 DEVCPU_GCB:SW_REGS:SEMA_INTR_ENA_SET

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 396 •* **Fields in SEMA_INTR_ENA_SET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SEMA_INTR_ENA_SET | 7:0 | One-shot | Set to set corresponding interrupt enable in SEMA_INTR_ENA. | 0x00 |

### 7.9.2.4 DEVCPU_GCB:SW_REGS:SEMA

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 8

*Table 397* • **Fields in SEMA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SEMA | 0 | R/W | General Semaphore.The process to read this field will read a '1' and thus be granted the semaphore. The semaphore is released by the interface by writing a '1' to this field.<br>Read :<br>'0': Semaphore was not granted.<br>'1': Semaphore was granted.<br><br>Write :<br>'0': No action.<br>'1': Release semaphore. | 0x1 |

### 7.9.2.5 DEVCPU_GCB:SW_REGS:SEMA_FREE

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 398* • **Fields in SEMA_FREE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SEMA_FREE | 7:0 | R/O | Show which semaphores that are currently free.<br>'0' : Corresponding semaphore is taken.<br>'1' : Corresponding semaphore is free. | 0xFF |

### 7.9.2.6 DEVCPU_GCB:SW_REGS:SW_INTR

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

This register provides a simple interface for interrupting on either software interrupt 0 or 1, without implementing semaphore support. Note: setting this field causes a short pulse on the corresponding interrupt connection, this kind of interrupt cannot be used in combination with the SW1_INTR_CONFIG.SW1_INTR_BYPASS feature.

*Table 399 •* **Fields in SW_INTR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW1_INTR | 1 | One-shot | Set this field to inject software interrupt 1. This field is automatically cleared after interrupt has been generated. | 0x0 |
| SW0_INTR | 0 | One-shot | Set this field to assert software interrupt 0. This field is automatically cleared after interrupt has been generated. | 0x0 |

### 7.9.2.7 DEVCPU_GCB:SW_REGS:MAILBOX

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 400 •* **Fields in MAILBOX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAILBOX | 31:0 | R/W | Read/write register. Atomic modifications can be performed by using the MAILBOX_CLR and MAILBOX_SET registers. | 0x00000000 |

### 7.9.2.8 DEVCPU_GCB:SW_REGS:MAILBOX_CLR

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 401 •* **Fields in MAILBOX_CLR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAILBOX_CLR | 31:0 | One-shot | Set bits in this register to atomically clear corresponding bits in the MAILBOX register. This register returns 0 on read. | 0x00000000 |

### 7.9.2.9 DEVCPU_GCB:SW_REGS:MAILBOX_SET

**Parent:** DEVCPU_GCB:SW_REGS

**Instances:** 1

*Table 402 •* **Fields in MAILBOX_SET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAILBOX_SET | 31:0 | One-shot | Set bits in this register to atomically set corresponding bits in the MAILBOX register. This register returns 0 on read. | 0x00000000 |

## 7.9.3 DEVCPU_GCB:VCORE_ACCESS

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 403 •* **Registers in VCORE_ACCESS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| VA_CTRL | 0x00000000 | 1 | Control register for VCore accesses | Page 348 |
| VA_ADDR | 0x00000004 | 1 | Address register for VCore accesses | Page 349 |
| VA_DATA | 0x00000008 | 1 | Data register for VCore accesses | Page 350 |
| VA_DATA_INCR | 0x0000000C | 1 | Data register for VCore accesses (w. auto increment of address) | Page 351 |
| VA_DATA_INERT | 0x00000010 | 1 | Data register for VCore accesses (will not initiate access) | Page 351 |

### 7.9.3.1 DEVCPU_GCB:VCORE_ACCESS:VA_CTRL

**Parent:** DEVCPU_GCB:VCORE_ACCESS

**Instances:** 1

*Table 404 •* **Fields in VA_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VA_ERR_RD | 3 | R/O | This field is set to the value of VA_CTRL:VA_ERR whenever one of the data registers ACC_DATA, ACC_DATA_INCR, or ACC_DATA_RO is read. By reading this field it is possible to determine if the last read-value from one of these registers was erred. | 0x0 |

*Table 404 •* **Fields in VA_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VA_ERR | 2 | R/O | This field is set if the access inside the VCore domain was terminated by an error. This situation can occur when accessing an unmapped part of the VCore memory-map or when accessing a target that reports error (e.g. accessing uninitialized DDR2 memory).<br>If an error occurs during reading, the read-data will be 0x80000000. So as an optimization, software only has to check for error if 0x80000000 is returned (and in that case VA_ERR_RD should be checked). When writing you should always check if successful. | 0x0 |
| VA_BUSY_RD | 1 | R/O | This field is set to the value of VA_CTRL:VA_BUSY whenever one of the data registers ACC_DATA, ACC_DATA_INCR, or ACC_DATA_RO is read. By reading this field it is possible to determine if the last read-value from one of these registers was valid. | 0x0 |
| VA_BUSY | 0 | R/O | This field is set by hardware when an access into VCore domain is started, and cleared when the access is done. | 0x0 |

### 7.9.3.2 DEVCPU_GCB:VCORE_ACCESS:VA_ADDR

**Parent:** DEVCPU_GCB:VCORE_ACCESS

**Instances:** 1

*Table 405 •* **Fields in VA_ADDR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VA_ADDR | 31:0 | R/W | The address to access in the VCore domain, all addresses must be 32-bit aligned (i.e. the two least significant bit must always be 0). When accesses are initiated using the ACC_DATA_INCR register, then this field is automatically incremented by 4 at the end of the transfer.<br>The memory region of the VCore that maps to switch-core registers may not be accessed by using these registers. | 0x00000000 |

### 7.9.3.3 DEVCPU_GCB:VCORE_ACCESS:VA_DATA

**Parent:** DEVCPU_GCB:VCORE_ACCESS

**Instances:** 1

The VA_DATA, VA_DATA_INCR, and VA_DATA_INERT registers are used for indirect access into the VCore domain. The functionality of the VA_DATA_INCR and VA_DATA_INERT registers are similar to this register - but with minor exceptions. These exceptions are fleshed out in the description of the respective registers.

***Table 406 •*** **Fields in VA_DATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VA_DATA | 31:0 | R/W | Reading or writing from/to this field initiates accesses into the VCore domain. While an access is ongoing (VA_CTRL:VA_BUSY is set) this field may not be written. It is possible to read this field while an access is ongoing, but the data returned will be 0x80000000.<br>When writing to this field; a write into the VCore domain is initiated to the address specified in the VA_ADDR register, with the data that was written to this field. Only 32-bit writes are supported. This field may not be written to until the VA_CTRL:VA_BUSY indicates that no accesses is ongoing.<br>When reading from this field; a read from the VCore domain is initiated from the address specified in the VA_ADDR register.<br>Important: The data that is returned from reading this field (and stating an access) is not the result of the newly initiated read, instead the data from the last access is returned. The result of the newly initiated read access will be ready once the VA_CTRL:VA_BUSY field shows that the access is done.<br>Note: When the result of a read-access is read from this field (the second read), a new access will automatically be initiated. This is desirable when reading a series of addresses from VCore domain. If a new access is not desirable, then the result should be read from the VA_DATA_INERT register instead of this field! | 0x00000000 |

### 7.9.3.4 DEVCPU_GCB:VCORE_ACCESS:VA_DATA_INCR

**Parent:** DEVCPU_GCB:VCORE_ACCESS

**Instances:** 1

*Table 407* • **Fields in VA_DATA_INCR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VA_DATA_INCR | 31:0 | R/W | This field behaves in the same way as ACC_DATA:ACC_DATA. Except when an access is initiated by using this field (either read or write); the address register (ACC_ADDR) is automatically incremented by 4 at the end of the access, i.e. when VA_CTRL:VA_BUSY is deasserted. | 0x00000000 |

### 7.9.3.5 DEVCPU_GCB:VCORE_ACCESS:VA_DATA_INERT

**Parent:** DEVCPU_GCB:VCORE_ACCESS

**Instances:** 1

*Table 408* • **Fields in VA_DATA_INERT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VA_DATA_INERT | 31:0 | R/W | This field behaves in the same way as ACC_DATA:ACC_DATA. Except accesses (read or write) does not initiate VCore accesses. Writing to this register just overwrites the value currently held by all of the data registers (ACC_DATA, ACC_DATA_INCR, and ACC_DATA_INERT). | 0x00000000 |

## 7.9.4 DEVCPU_GCB:GPIO

**Parent:** DEVCPU_GCB

**Instances:** 1

General Purpose I/O Control configuration and status registers.

Each register in this group contains one field with one bit per GPIO pin. Bit 0 in each field corresponds to GPIO0, bit 1 to GPIO1, and so on.

*Table 409* • **Registers in GPIO**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| GPIO_OUT_SET | 0x00000000 | 1 | GPIO output set | Page 352 |

*Table 409 •* **Registers in GPIO *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| GPIO_OUT_CLR | 0x00000004 | 1 | GPIO output clear | Page 352 |
| GPIO_OUT | 0x00000008 | 1 | GPIO output | Page 352 |
| GPIO_IN | 0x0000000C | 1 | GPIO input | Page 353 |
| GPIO_OE | 0x00000010 | 1 | GPIO pin direction | Page 353 |
| GPIO_INTR | 0x00000014 | 1 | GPIO interrupt | Page 353 |
| GPIO_INTR_ENA | 0x00000018 | 1 | GPIO interrupt enable | Page 354 |
| GPIO_INTR_IDENT | 0x0000001C | 1 | GPIO interrupt identity | Page 354 |
| GPIO_ALT | 0x00000020 | 2<br>0x00000004 | GPIO alternate functions | Page 354 |

### 7.9.4.1 DEVCPU_GCB:GPIO:GPIO_OUT_SET

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

*Table 410 •* **Fields in GPIO_OUT_SET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_OUT_SET | 31:0 | One-shot | Setting a bit in this field will immediately set the corresponding bit in GPIO_O::G_OUT. Reading this register always return 0.<br>'0': No change<br>'1': Corresponding bit in GPIO_O::OUT is set. | 0x00000000 |

### 7.9.4.2 DEVCPU_GCB:GPIO:GPIO_OUT_CLR

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

*Table 411 •* **Fields in GPIO_OUT_CLR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_OUT_CLR | 31:0 | One-shot | Setting a bit in this field will immediately clear the corresponding bit in GPIO_O::G_OUT. Reading this register always return 0.<br>'0': No change<br>'1': Corresponding bit in GPIO_O::OUT is cleared. | 0x00000000 |

### 7.9.4.3 DEVCPU_GCB:GPIO:GPIO_OUT

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

In a multi-threaded software environment using the registers GPIO_OUT_SET and GPIO_OUT_CLR for modifying GPIO values removes the need for software-locked access.

***Table 412 •*** **Fields in GPIO_OUT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_OUT | 31:0 | R/W | Controls the value on the GPIO pins enabled for output (via the GPIO_OE register). This field can be modified directly or by using the GPIO_O_SET and GPIO_O_CLR registers. | 0x00000000 |

### 7.9.4.4 DEVCPU_GCB:GPIO:GPIO_IN

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

***Table 413 •*** **Fields in GPIO_IN**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_IN | 31:0 | R/O | GPIO input register. Reflects the current state of the corresponding GPIO pins. | 0x00000000 |

### 7.9.4.5 DEVCPU_GCB:GPIO:GPIO_OE

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

***Table 414 •*** **Fields in GPIO_OE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_OE | 31:0 | R/W | Configures the direction of the GPIO pins.<br>'0': Input<br>'1': Output | 0x00000000 |

### 7.9.4.6 DEVCPU_GCB:GPIO:GPIO_INTR

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

*Table 415* • **Fields in GPIO_INTR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_INTR | 31:0 | Sticky | Indicates whether a GPIO input has changed since last clear.<br>'0': No change<br>'1': GPIO has changed | 0x00000000 |

### 7.9.4.7 DEVCPU_GCB:GPIO:GPIO_INTR_ENA

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

*Table 416* • **Fields in GPIO_INTR_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_INTR_ENA | 31:0 | R/W | Enables individual GPIO pins for interrupt. | 0x00000000 |

### 7.9.4.8 DEVCPU_GCB:GPIO:GPIO_INTR_IDENT

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 1

*Table 417* • **Fields in GPIO_INTR_IDENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_INTR_IDENT | 31:0 | R/O | Shows which GPIO sources that are currently interrupting. This field is the result of an AND-operation between the GPIO_INTR and the GPIO_INTR_ENA registers. | 0x00000000 |

### 7.9.4.9 DEVCPU_GCB:GPIO:GPIO_ALT

**Parent:** DEVCPU_GCB:GPIO

**Instances:** 2

*Table 418 •* **Fields in GPIO_ALT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| G_ALT | 31:0 | R/W | Configures alternate functions for individual GPIO bits. This field is replicated two times, the functionality of the GPIO is determined by the bit in this field corresponding to the GPIO for BOTH replications.<br>For example, to enable alternate function 1 for GPIO number 3; then bit 3 in G_ALT[0] must be set and bit 3 in G_ALT[1] must be cleared.<br>The encoding describes the result of setting bits in both replications of this field per GPIO. That is, the encoding shows the following concatenation "G_ALT[1] & G_ALT[0]" per GPIO.<br>"00": GPIO mode<br>"01": Alternate mode 1<br>"10": Alternate mode 2<br>"11": Reserved | 0x00000000 |

## 7.9.5 DEVCPU_GCB:DEVCPU_RST_REGS

**Parent:** DEVCPU_GCB

**Instances:** 1

Resets the chip

*Table 419 •* **Registers in DEVCPU_RST_REGS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SOFT_CHIP_RST | 0x00000000 | 1 | Reset part or the whole chip | Page 355 |
| SOFT_DEVCPU_RST | 0x00000004 | 1 | Soft reset of devcpu. | Page 356 |

### 7.9.5.1 DEVCPU_GCB:DEVCPU_RST_REGS:SOFT_CHIP_RST

**Parent:** DEVCPU_GCB:DEVCPU_RST_REGS

**Instances:** 1

*Table 420 •* **Fields in SOFT_CHIP_RST**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SOFT_PHY_RST | 1 | R/W | Clear this field to release reset in the Cu-PHY. This field is automatically set during hard-reset and soft-reset of the chip. After reset is released the PHY will indicate when it is ready to be accessed via DEVCPU_GCB::MISC_STAT.PHY_READY. | 0x1 |
| SOFT_CHIP_RST | 0 | R/W | Set this field to reset the whole chip. This field is automatically cleared by the reset.<br>Note: It is possible for the VCore to protect itself from soft-reset of the chip, for more info see RESET.CORE_RST_PROTECT inside the VCore register space. | 0x0 |

### 7.9.5.2 DEVCPU_GCB:DEVCPU_RST_REGS:SOFT_DEVCPU_RST

**Parent:** DEVCPU_GCB:DEVCPU_RST_REGS

**Instances:** 1

*Table 421 •* **Fields in SOFT_DEVCPU_RST**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SOFT_XTR_RST | 1 | R/W | Set this field to reset the extraction logic. The reset remains asserted until this field is cleared.<br>Note: Extraction logic is also reset while SOFT_CHIP_RST.SOFT_NON_CFG_RST is set. | 0x0 |
| SOFT_INJ_RST | 0 | R/W | Set this field to reset the injection logic. The reset remains asserted until this field is cleared.<br>Note: Injection logic is also reset while SOFT_CHIP_RST.SOFT_NON_CFG_RST is set. | 0x0 |

## 7.9.6 DEVCPU_GCB:MIIM

**Parent:** DEVCPU_GCB

**Instances:** 2

*Table 422* • **Registers in MIIM**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MII_STATUS | 0x00000000 | 1 | MIIM Status | Page 357 |
| MII_CMD | 0x00000008 | 1 | MIIM Command | Page 358 |
| MII_DATA | 0x0000000C | 1 | MIIM Reply Data | Page 359 |
| MII_CFG | 0x00000010 | 1 | MIIM Configuration | Page 359 |
| MII_SCAN_0 | 0x00000014 | 1 | MIIM Scan 0 | Page 360 |
| MII_SCAN_1 | 0x00000018 | 1 | MIIM Scan 1 | Page 360 |
| MII_SCAN_LAST_RSLTS | 0x0000001C | 1 | MIIM Results | Page 360 |
| MII_SCAN_LAST_RSLTS_VLD | 0x00000020 | 1 | MIIM Results | Page 361 |

### 7.9.6.1 DEVCPU_GCB:MIIM:MII_STATUS

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 423* • **Fields in MII_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_STAT_BUSY | 3 | R/O | Indicates the current state of the MIIM controller. When read operations are done (no longer busy), then read data is available via the DEVCPU_GCB::MII_DATA register.<br>0: MIIM controller is in idle state<br>1: MIIM controller is busy performing MIIM cmd (Either read or read cmd). | 0x0 |
| MIIM_STAT_OPR_PEND | 2 | R/O | The MIIM controller has a CMD fifo of depth one. When this field is 0, then it is safe to write another MIIM command to the MIIM controller.<br>0 : Read or write not pending<br>1 : Read or write pending. | 0x0 |
| MIIM_STAT_PENDING_RD | 1 | R/O | Indicates whether a read operation via the MIIM interface is in progress or not.<br>0 : Read not in progress<br>1 : Read in progress. | 0x0 |

*Table 423 •* **Fields in MII_STATUS *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_STAT_PENDING_WR | 0 | R/O | Indicates whether a write operation via the MIIM interface is in progress or not.<br>0 : Write not in progress<br>1 : Write in progress. | 0x0 |
| MIIM_SCAN_COMPLETE | 4 | R/O | Signals if all PHYs have been scanned ( with auto scan ) at least once.<br>0 : Auto scan has not scanned all PHYs.<br>1 : Auto scan has scanned all PHY at least once. | 0x0 |

### 7.9.6.2 DEVCPU_GCB:MIIM:MII_CMD

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 424 •* **Fields in MII_CMD**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_CMD_VLD | 31 | One-shot | Must be set for starting a new PHY access. This bit is automatically cleared.<br>0 : Write to this register is ignored.<br>1 : Write to this register is processed. | 0x0 |
| MIIM_CMD_PHYAD | 29:25 | R/W | Indicates the addressed PHY number. | 0x00 |
| MIIM_CMD_REGAD | 24:20 | R/W | Indicates the addressed of the register within the PHY that shall be accessed. | 0x00 |
| MIIM_CMD_WRDATA | 19:4 | R/W | Data to be written in the PHY register. | 0x0000 |
| MIIM_CMD_SINGLE_SCAN | 3 | R/W | Select if scanning of the PHY shall be done once, or scanning should be done continuously.<br>0 : Do continuously PHY scanning<br>1 : Stop once all PHY have been scanned. | 0x0 |

*Table 424 •* **Fields in MII_CMD *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_CMD_OPR_FIELD | 2:1 | R/W | Indicates type of operation.<br>Clause 22:<br><br>01 : Write<br>10 : Read<br><br>Clause 45:<br><br>00 : Address<br>01 : Write<br>10 : Read inc.<br>11 : Read. | 0x0 |
| MIIM_CMD_SCAN | 0 | R/W | Indicates whether automatic scanning of PHY registers is enabled. When enabled, the PHY-number for each automatic read is continuously round-robined from PHY_ADDR_LOW through PHY_ADDR_HIGH. This function is started upon a read operation (ACCESS_TYPE).<br>Scan MUST be disabled when doing any configuration of the MIIM controller.<br>0 : Disabled<br>1 : Enabled. | 0x0 |

### 7.9.6.3 DEVCPU_GCB:MIIM:MII_DATA

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 425 •* **Fields in MII_DATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_DATA_SUCCESS | 17:16 | R/O | Indicates whether a read operation failed or succeeded.<br>00 : OK<br>11 : Error | 0x0 |
| MIIM_DATA_RDDATA | 15:0 | R/O | Data read from PHY register. | 0x0000 |

### 7.9.6.4 DEVCPU_GCB:MIIM:MII_CFG

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 426* • **Fields in MII_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_CFG_PRESCALE | 7:0 | R/W | Configures the MIIM clock frequency. This is computed as system_clk/(2*(1+X)), where X is the value written to this register. Note : Setting X to 0 is invalid and will result in the same frequency as setting X to 1. | 0x32 |
| MIIM_ST_CFG_FIELD | 10:9 | R/W | The ST (start-of-frame) field of the MIIM frame format adopts the value of this field. This must be configured for either clause 22 or 45 MIIM operation.<br>"01": Clause 22<br>"00": Clause 45<br>Other values are reserved. | 0x1 |

### 7.9.6.5 DEVCPU_GCB:MIIM:MII_SCAN_0

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 427* • **Fields in MII_SCAN_0**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_SCAN_PHYADHI | 9:5 | R/W | Indicates the high PHY number to scan during automatic scanning. | 0x00 |
| MIIM_SCAN_PHYADLO | 4:0 | R/W | Indicates the low PHY number to scan during automatic scanning. | 0x00 |

### 7.9.6.6 DEVCPU_GCB:MIIM:MII_SCAN_1

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 428* • **Fields in MII_SCAN_1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_SCAN_MASK | 31:16 | R/W | Indicates the mask for comparing the PHY registers during automatic scan. | 0x0000 |
| MIIM_SCAN_EXPECT | 15:0 | R/W | Indicates the expected value for comparing the PHY registers during automatic scan. | 0x0000 |

### 7.9.6.7 DEVCPU_GCB:MIIM:MII_SCAN_LAST_RSLTS

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 429 •* **Fields in MII_SCAN_LAST_RSLTS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_LAST_RSLT | 31:0 | R/O | Indicates for each PHY if a PHY register has matched the expected value (with mask).<br>This register reflects the value of the last reading of the phy register.<br>0 : Mismatch.<br>1 : Match. | 0x00000000 |

#### 7.9.6.8 DEVCPU_GCB:MIIM:MII_SCAN_LAST_RSLTS_VLD

**Parent:** DEVCPU_GCB:MIIM

**Instances:** 1

*Table 430 •* **Fields in MII_SCAN_LAST_RSLTS_VLD**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_LAST_RSLT_VLD | 31:0 | R/O | Indicates for each PHY if a PHY register matched are valid or not.<br>0 : Scan result not valid.<br>1 : Scan result valid. | 0x00000000 |

### 7.9.7 DEVCPU_GCB:MIIM_READ_SCAN

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 431 •* **Registers in MIIM_READ_SCAN**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MII_SCAN_RSLTS_STICKY | 0x00000000 | 2<br>0x00000004 | MIIM Results | Page 361 |

#### 7.9.7.1 DEVCPU_GCB:MIIM_READ_SCAN:MII_SCAN_RSLTS_STICKY

**Parent:** DEVCPU_GCB:MIIM_READ_SCAN

**Instances:** 2

*Table 432 •* **Fields in MII_SCAN_RSLTS_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM_SCAN_RSLTS_STICKY | 31:0 | R/O | Indicates for each PHY if a PHY register has had a mismatch of the expected value (with mask) since last reading of MIIM_SCAN_RSLTS_STICKY.<br><br>Result is sticky, and result will indicate if there has been a mismatch since the last reading of this register.<br><br>Upon reading this register, all bits are reset to '1'.<br>0 : Mismatch<br>1 : Match. | 0x00000000 |

## 7.9.8 DEVCPU_GCB:RAM_STAT

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 433 •* **Registers in RAM_STAT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| RAM_INTEGRITY_ERR_STICKY | 0x00000000 | 1 | QS RAM status | Page 362 |

### 7.9.8.1 DEVCPU_GCB:RAM_STAT:RAM_INTEGRITY_ERR_STICKY

**Parent:** DEVCPU_GCB:RAM_STAT

**Instances:** 1

*Table 434 •* **Fields in RAM_INTEGRITY_ERR_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| QS_XTR_RAM_INTGR_ERR_STICKY | 0 | Sticky | Integrity error for QS_XTR RAM<br>'0': No RAM integrity check error occurred<br>'1': A RAM integrity check error occurred<br>Bit is cleared by writing a '1' to this position. | 0x0 |

## 7.9.9 DEVCPU_GCB:MISC

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 435 •* **Registers in MISC**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MISC_CFG | 0x00000000 | 1 | Miscellaneous Configuration Register | Page 363 |
| MISC_STAT | 0x00000004 | 1 | | Page 363 |
| PHY_SPEED_1000_STAT | 0x00000008 | 1 | | Page 364 |
| PHY_SPEED_100_STAT | 0x0000000C | 1 | | Page 364 |
| PHY_SPEED_10_STAT | 0x00000010 | 1 | | Page 364 |
| DUPLEXC_PORT_STAT | 0x00000014 | 1 | | Page 364 |

### 7.9.9.1 DEVCPU_GCB:MISC:MISC_CFG

**Parent:** DEVCPU_GCB:MISC

**Instances:** 1

Register to control various muxing in the IO-ring.

*Table 436 •* **Fields in MISC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW_MODE | 7:6 | R/W | Set the sw_mode for HSIO.<br>0: Use for VSC7425-02 (12x CuPHY + 1x QSGMII + 2x SGMII), VSC7426-02, and VSC7427-02.<br>1: Reserved.<br>2: Use for VSC7424-02 and VSC7425-02 (12x CuPHY + 6x SGMII).<br>3: Reserved. | 0x0 |
| QSGMII_FLIP_LANE1 | 5 | R/W | Flip or swap lanes in QSGMII#1. | 0x0 |
| QSGMII_FLIP_LANE2 | 4 | R/W | Flip or swap lanes in QSGMII#2. | 0x0 |
| QSGMII_FLIP_LANE3 | 3 | R/W | Flip or swap lanes in QSGMII#3. | 0x0 |
| QSGMII_SHYST_DIS | 2 | R/W | Disable hysteresis of synchronization state machine. | 0x0 |
| QSGMII_E_DET_ENA | 1 | R/W | Enable 8b10b error propagation (8b10b error code-groups are replaced by K70.7 error symbols. | 0x0 |
| QSGMII_USE_I1_ENA | 0 | R/W | Use I1 during idle sequencing only. | 0x0 |

### 7.9.9.2 DEVCPU_GCB:MISC:MISC_STAT

**Parent:** DEVCPU_GCB:MISC

**Instances:** 1

*Table 437* • **Fields in MISC_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PHY_READY | 3 | R/O | This field is set high when the PHY is ready for access after release of PHY reset via DEVCPU_GCB::SOFT_CHIP_RST.SOFT_PHY_RST. | 0x0 |

### 7.9.9.3 DEVCPU_GCB:MISC:PHY_SPEED_1000_STAT

**Parent:** DEVCPU_GCB:MISC

**Instances:** 1

*Table 438* • **Fields in PHY_SPEED_1000_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SPEED_1000 | 11:0 | R/O | p2m_speed1000c status from PHY | 0x000 |

### 7.9.9.4 DEVCPU_GCB:MISC:PHY_SPEED_100_STAT

**Parent:** DEVCPU_GCB:MISC

**Instances:** 1

*Table 439* • **Fields in PHY_SPEED_100_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SPEED_100 | 11:0 | R/O | p2m_speed100 status from PHY | 0x000 |

### 7.9.9.5 DEVCPU_GCB:MISC:PHY_SPEED_10_STAT

**Parent:** DEVCPU_GCB:MISC

**Instances:** 1

*Table 440* • **Fields in PHY_SPEED_10_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SPEED_10 | 11:0 | R/O | p2m_speed10 status from PHY | 0x000 |

### 7.9.9.6 DEVCPU_GCB:MISC:DUPLEXC_PORT_STAT

**Parent:** DEVCPU_GCB:MISC

**Instances:** 1

*Table 441* • **Fields in DUPLEXC_PORT_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DUPLEXC | 11:0 | R/O | p2m_duplexc_port status from PHY | 0x000 |

## 7.9.10 DEVCPU_GCB:SIO_CTRL

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 442* • **Registers in SIO_CTRL**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SIO_INPUT_DATA | 0x00000000 | 4<br>0x00000004 | Input data registers | Page 365 |
| SIO_INT_POL | 0x00000010 | 4<br>0x00000004 | Interrupt polarity for each GPIO | Page 366 |
| SIO_PORT_INT_ENA | 0x00000020 | 1 | Interrupt enable register for each port. | Page 366 |
| SIO_PORT_CONFIG | 0x00000024 | 32<br>0x00000004 | Configuration of output data values | Page 366 |
| SIO_PORT_ENABLE | 0x000000A4 | 1 | Port enable register | Page 367 |
| SIO_CONFIG | 0x000000A8 | 1 | General configuration register | Page 367 |
| SIO_CLOCK | 0x000000AC | 1 | Configuration of the serial IO clock frequency | Page 369 |
| SIO_INT_REG | 0x000000B0 | 4<br>0x00000004 | Interrupt register | Page 369 |

### 7.9.10.1 DEVCPU_GCB:SIO_CTRL:SIO_INPUT_DATA

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 4

*Table 443* • **Fields in SIO_INPUT_DATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| S_IN | 31:0 | R/O | Serial input data. The first replication holds bit 0 from all ports, the 2nd replication holds bit 1 from all ports, etc.<br>Values of disabled gpios are undefined.<br>bit order: (port-31 bit-n down to port-0 bit-n) | 0x00000000 |

### 7.9.10.2 DEVCPU_GCB:SIO_CTRL:SIO_INT_POL

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 4

*Table 444* • **Fields in SIO_INT_POL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_POL | 31:0 | R/W | Interrupt polarity. Bit n from all ports.<br>This register defines at which logic value an interrupt is generated.<br>For bit 0, this register is also used to define the polarity of the "loss of signal" output.<br>0 : interrupt at logic value '1'<br>1 : interrupt at logic value '0'<br>For "loss of signal":<br>0 : "loss of signal" is active high<br>1: "loss of signal" is active low | 0x00000000 |

### 7.9.10.3 DEVCPU_GCB:SIO_CTRL:SIO_PORT_INT_ENA

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 1

*Table 445* • **Fields in SIO_PORT_INT_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_ENA | 31:0 | R/W | Interrupt enable vector with one enable bit for each port.<br><br>0 : Interrupt is disabled for the port.<br>1 : Interrupt is enabled for the port.<br>port order: (portN down to port0) | 0x00000000 |

### 7.9.10.4 DEVCPU_GCB:SIO_CTRL:SIO_PORT_CONFIG

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 32

*Table 446* • **Fields in SIO_PORT_CONFIG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BIT_SOURCE | 11:0 | R/W | Output source select for the four outputs from each port.<br>The source select is encoded using three bits for each output bit.<br>The placement of the source select bits for each output bit in the register:<br>Output bit 0: (2 down to 0)<br>Output bit 1: (5 down to 3)<br>Output bit 2: (8 down to 6)<br>Output bit 3: (11 down to 9)<br>Source select encoding for each output bit:<br>0 : Forced '0'<br>1 : Forced '1'<br>2 : Blink mode 0<br>3 : Blink mode 1<br>4 : Link activity blink mode 0<br>5 : Link activity blink mode 1<br>6 : Link activity blink mode 0 inversed polarity<br>7 : Link activity blink mode 1 inversed polarity | 0x000 |

### 7.9.10.5 DEVCPU_GCB:SIO_CTRL:SIO_PORT_ENABLE

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 1

*Table 447* • **Fields in SIO_PORT_ENABLE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| P_ENA | 31:0 | R/W | Port enable vector with one enable bit for each port.<br>0 : Port is disabled.<br>1 : Port is enabled.<br>Port order: (portN down to port0) | 0x00000000 |

### 7.9.10.6 DEVCPU_GCB:SIO_CTRL:SIO_CONFIG

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 1

*Table 448* • **Fields in SIO_CONFIG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SIO_BMODE_1 | 21:20 | R/W | Configuration for blink mode 1. Supports three different blink modes and a "burst toggle" mode in which blink mode 1 will alternate for each burst.<br>0 : Blink freq approximately 20Hz<br>1 : Blink freq approximately 10Hz.<br>2 : Blink freq approximately 5Hz.<br>3 : Burst toggle. | 0x0 |
| SIO_BMODE_0 | 19:18 | R/W | Configuration of blink mode 0. Supports four different blink modes.<br>Configuration of blink mode 0. Supports four different blink modes.<br>0 : Blink freq approximately 20Hz<br>1 : Blink freq approximately 10Hz.<br>2 : Blink freq approximately 5Hz.<br>3 : Blink freq approximately 2.5Hz. | 0x0 |
| SIO_BLINK_RESET | 17 | R/W | Reset the blink counters. Used to synchronize the blink modes between different chips.<br>0 : Blink counter is running.<br>1 : Blink counter is reset until sio_blink_reset is unset again. | 0x0 |
| SIO_INT_ENA | 16:13 | R/W | Bit interrupt enable. Enables interrupts for the four gpios in a port. Is applied to all ports.<br>0: Interrupt is disabled for bit n for all ports.<br>1: Interrupt is enabled for bit n for all ports. | 0x0 |
| SIO_BURST_GAP_DIS | 12 | R/W | Set to disable burst gap. | 0x0 |
| SIO_BURST_GAP | 11:7 | R/W | Configures the length of burst gap in steps of approx. 1 ms. Burst gap can be disabled by setting SIO_CONFIG.SIO_BURST_GAP_DIS.<br>0: 1.05 ms burst gap.<br>1: 2.10 ms burst gap.<br>31: 33.55 ms burst gap. | 0x00 |
| SIO_SINGLE_SHOT | 6 | One-shot | Use this to output a single burst. Will be cleared by hardware when the burst has finished. | 0x0 |
| SIO_AUTO_REPEAT | 5 | R/W | Use this to output repeated bursts interleaved with burst gaps. Must be manually reset again to stop output of bursts. | 0x0 |
| SIO_LD_POLARITY | 4 | R/W | Polarity of the "Ld" signal<br>0: load signal is active low<br>1: load signal is active high | 0x0 |

*Table 448 •* **Fields in SIO_CONFIG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SIO_PORT_WIDTH | 3:2 | R/W | Number of gpios pr. port.<br>0: 1 gpio pr. port.<br>1: 2 gpios pr. port.<br>2: 3 gpios pr. port.<br>3: 4 gpios pr. port. | 0x0 |
| SIO_REVERSE_OUTPUT | 1 | R/W | Reverse the output bitstream.<br><br>The default order of the output bit stream is (displayed in transmitted order):<br>(portN bit3, portN bit2, ...., port0 bit1, port0 bit0)<br><br>The reverse order of the output bit stream is (displayed in transmitted order):<br>(port0 bit0, port0 bit1, ...., portN bit2, portN bit3)<br>0 : Do not reverse.<br>1 : Reverse. | 0x0 |
| SIO_REVERSE_INPUT | 0 | R/W | Reverse the input bitstream.<br>The default order of the input bit stream is (displayed in received order):<br>(port0 bit0, port0 bit1, ...., portN bit2, portN bit3)<br>The reverse order of the input bit stream is (displayed in received order):<br>(portN bit3, portN bit2, ...., port0 bit1, port0 bit0)<br>0: Do not reverse.<br>1: Reverse. | 0x0 |

### 7.9.10.7 DEVCPU_GCB:SIO_CTRL:SIO_CLOCK

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 1

*Table 449 •* **Fields in SIO_CLOCK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SIO_CLK_FREQ | 11:0 | R/W | SIO controller clock frequency.<br>Divides the 250MHz system clk with value of this field. E.g. the system clk is 250 MHz and this field is set to 10, the output frequency will be 25 MHz.<br>0 : Disable clock.<br>1 : Reserved, do not use.<br>Others : Clock divider value. | 0x000 |

### 7.9.10.8 DEVCPU_GCB:SIO_CTRL:SIO_INT_REG

**Parent:** DEVCPU_GCB:SIO_CTRL

**Instances:** 4

*Table 450* • **Fields in SIO_INT_REG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_REG | 31:0 | Sticky | Interrupt register. Bit n from all ports. Disabled gpios are always '0'.<br>0: No interrupt for given gpio.<br>1: Interrupt for given gpio.<br>bit order (portM bit-n down to portM bit-0). | 0x00000000 |

## 7.9.11 DEVCPU_GCB:FAN_CFG

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 451* • **Registers in FAN_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| FAN_CFG | 0x00000000 | 1 | Configuration register for the fan controller | Page 370 |

### 7.9.11.1 DEVCPU_GCB:FAN_CFG:FAN_CFG

**Parent:** DEVCPU_GCB:FAN_CFG

**Instances:** 1

*Table 452* • **Fields in FAN_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PWM_FREQ | 5:3 | R/W | Set the frequency of the PWM output<br><br>0: 25 kHz<br>1: 120 Hz<br>2: 100 Hz<br>3: 80 Hz<br>4: 60 Hz<br>5: 40 Hz<br>6: 20 Hz<br>7: 10 Hz | 0x0 |
| INV_POL | 2 | R/W | Define the polarity of the PWM output.<br>0: PWM is logic 1 when "on"<br>1: PWM is logic 0 when "on" | 0x0 |

*Table 452 •* **Fields in FAN_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GATE_ENA | 1 | R/W | Enable gating of the TACH input by the PWM output so that only TACH pulses received when PWM is "on" are counted.<br>0: Disabled<br>1: Enabled | 0x0 |
| PWM_OPEN_COL_ENA | 0 | R/W | Configure the PWM output to be open collector | 0x0 |
| DUTY_CYCLE | 23:16 | R/W | Define the duty cycle<br>0x00: Always "off"<br>0xFF: Always "on" | 0x00 |

## 7.9.12 DEVCPU_GCB:FAN_STAT

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 453 •* **Registers in FAN_STAT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| FAN_CNT | 0x00000000 | 1 | TACH counter | Page 371 |

### 7.9.12.1 DEVCPU_GCB:FAN_STAT:FAN_CNT

**Parent:** DEVCPU_GCB:FAN_STAT

**Instances:** 1

*Table 454 •* **Fields in FAN_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FAN_CNT | 15:0 | R/O | Counts the number of rising edges on the TACH input. The counter is wrapping. | 0x0000 |

## 7.9.13 DEVCPU_GCB:PTP_CFG

**Parent:** DEVCPU_GCB

**Instances:** 1

Configuration registers for PTP

*Table 455* • **Registers in PTP_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PTP_MISC_CFG | 0x00000000 | 1 | Misc Configuration Register for PTP | Page 372 |
| PTP_UPPER_LIMIT_CFG | 0x00000004 | 1 | Configuration register for master counter upper limit | Page 372 |
| PTP_UPPER_LIMIT_1_TIME_ADJ_CFG | 0x00000008 | 1 | Configuration register for master counter upper limit one time adjustment | Page 372 |
| PTP_SYNC_INTR_ENA_CFG | 0x0000000C | 1 | Sync Interrupt enable register | Page 373 |
| CLK_ADJ_CFG | 0x0000001C | 1 | Configuration register for generated clock frequency adjustment | Page 373 |

### 7.9.13.1 DEVCPU_GCB:PTP_CFG:PTP_MISC_CFG

**Parent:** DEVCPU_GCB:PTP_CFG

**Instances:** 1

Misc Configuration Register for PTP

*Table 456* • **Fields in PTP_MISC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_ENA | 0 | R/W | Enable master counter.<br>0: Master counter disabled.<br>1: Master counter enabled. | 0x0 |

### 7.9.13.2 DEVCPU_GCB:PTP_CFG:PTP_UPPER_LIMIT_CFG

**Parent:** DEVCPU_GCB:PTP_CFG

**Instances:** 1

Configuration register for master counter upper limit

*Table 457* • **Fields in PTP_UPPER_LIMIT_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_UPPER_LIMIT | 27:0 | R/W | Counter value where the Master counter should be reset<br>Units is time in clock_ticks.<br>1 clock tick is 4 ns, if system_clk is set to 250MHz. | 0xEE6B27F |

### 7.9.13.3 DEVCPU_GCB:PTP_CFG:PTP_UPPER_LIMIT_1_TIME_ADJ_CFG

**Parent:** DEVCPU_GCB:PTP_CFG

**Instances:** 1

Configuration register for master counter upper limit one time adjustment

***Table 458 •*** **Fields in PTP_UPPER_LIMIT_1_TIME_ADJ_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_UPPER_LIMIT_1_TIME_ADJ_SHOT | 31 | One-shot | One time enable for PTP_UPPER_LIMIT_1_TIME_ADJ<br>0: Normal operation<br>1: Timer is adjusted by usage of PTP_UPPER_LIMIT_1_TIME_ADJ<br>Bit is cleared by HW | 0x0 |
| PTP_UPPER_LIMIT_1_TIME_ADJ | 27:0 | R/W | Counter value where the Master counter should be reset<br>Units is time in clock_ticks.<br>1 clock tick is 4 ns | 0xEE6B27F |

### 7.9.13.4 DEVCPU_GCB:PTP_CFG:PTP_SYNC_INTR_ENA_CFG

**Parent:** DEVCPU_GCB:PTP_CFG

**Instances:** 1

Sync Interrupt enable register

***Table 459 •*** **Fields in PTP_SYNC_INTR_ENA_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SYNC_STAT_ENA | 0 | R/W | Interrupt mask. Masks interrupt generation when Master Timer generates a synchronization pulse.<br>'0': Interrupt is not generated<br>'1': Interrupt is generated | 0x0 |

### 7.9.13.5 DEVCPU_GCB:PTP_CFG:CLK_ADJ_CFG

**Parent:** DEVCPU_GCB:PTP_CFG

**Instances:** 1

Configuration register for generated clock frequency adjustment

*Table 460 •* **Fields in CLK_ADJ_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLK_ADJ_DIR | 31 | R/W | Clock frequency adjustment direction.<br>0: Positive adjustment. Every N cycles a 1 is added to the counter. => clock period is decrease, clock frequency is increased.<br>1: Negative adjustment. Every N cycles a 1 is subtracted from the counter. => clock period is increase, clock frequency is decreased. | 0x0 |
| CLK_ADJ_ENA | 30 | R/W | Clock frequency adjust enable.<br>0: Adjustment Disabled<br>1: Adjustment Enabled | 0x0 |
| CLK_ADJ_UPD | 29 | R/W | Defines when the updated adjustment value and direction takes effect.<br>0: updated values take immediate effect.<br>1: updated values take effect after the next sync pulse. | 0x0 |
| CLK_ADJ | 27:0 | R/W | Clock frequency adjust.<br>N: Number of clock cycles after which the counter for the clock must be adjusted. | 0x0004E1F |

## 7.9.14 DEVCPU_GCB:PTP_STAT

**Parent:** DEVCPU_GCB

**Instances:** 1

Status registers for PTP

*Table 461 •* **Registers in PTP_STAT**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PTP_CURRENT_TIME_STAT | 0x00000000 | 1 | Current PTP master timer value | Page 374 |
| PTP_EVT_STAT | 0x00000008 | 1 | Stick register for external sync current time status | Page 375 |

### 7.9.14.1 DEVCPU_GCB:PTP_STAT:PTP_CURRENT_TIME_STAT

**Parent:** DEVCPU_GCB:PTP_STAT

**Instances:** 1

Current PTP master timer value

*Table 462 •* **Fields in PTP_CURRENT_TIME_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_CURRENT_TIME | 27:0 | R/O | Current master counter value. Unit is 4 ns. | 0x0000000 |

### 7.9.14.2 DEVCPU_GCB:PTP_STAT:PTP_EVT_STAT

**Parent:** DEVCPU_GCB:PTP_STAT

**Instances:** 1

Stick register for external sync current time status

*Table 463 •* **Fields in PTP_EVT_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLK_ADJ_UPD_STICKY | 2 | Sticky | Identifies if the adjust value update has already happened in case the adjustment is only allowed to take place at sync.<br>If update is allowed to take place immediately the sticky bit is unused.<br>0: updated has not yet happened<br>1: updated has happened<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| SYNC_STAT | 0 | Sticky | Master timer has generated a synchronization pulse to the Slave Timers.<br>'0': No master timer wrap happened.<br>'1': Master timer wrap happened.<br>Bit is cleared by writing a '1' to this position. | 0x0 |

## 7.9.15 DEVCPU_GCB:PTP_TIMERS

**Parent:** DEVCPU_GCB

**Instances:** 1

*Table 464 •* **Registers in PTP_TIMERS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PTP_TOD_SECS | 0x00000000 | 1 | Time of day (Seconds) | Page 376 |
| PTP_TOD_NANOSECS | 0x00000004 | 1 | Time of day (Nanoseconds) | Page 376 |
| PTP_DELAY | 0x00000008 | 1 | Delay timer | Page 376 |

*Table 464 •* **Registers in PTP_TIMERS *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PTP_TIMER_CTRL | 0x0000000C | 1 | Control register for PTP timers | Page 377 |

### 7.9.15.1 DEVCPU_GCB:PTP_TIMERS:PTP_TOD_SECS

**Parent:** DEVCPU_GCB:PTP_TIMERS

**Instances:** 1

Time of day (Seconds)

*Table 465 •* **Fields in PTP_TOD_SECS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_TOD_SECS | 31:0 | R/O | Seconds fraction of time of day timer at latch time (PTP_TIMER_CTRL.PTP_LATCH ). Unit is seconds. | 0x00000000 |

### 7.9.15.2 DEVCPU_GCB:PTP_TIMERS:PTP_TOD_NANOSECS

**Parent:** DEVCPU_GCB:PTP_TIMERS

**Instances:** 1

Time of day (Nanoseconds)

*Table 466 •* **Fields in PTP_TOD_NANOSECS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_TOD_NANOSECS | 27:0 | R/O | Nanoseconds fraction of time of day timer at latch time (PTP_TIMER_CTRL.PTP_LATCH ). Unit is 4 ns. | 0x0000000 |

### 7.9.15.3 DEVCPU_GCB:PTP_TIMERS:PTP_DELAY

**Parent:** DEVCPU_GCB:PTP_TIMERS

**Instances:** 1

*Table 467 •* **Fields in PTP_DELAY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_DELAY | 31:0 | R/O | Delay timer in Rx/Tx timestampers at latch time (PTP_TIMER_CTRL.PTP_LATCH ). Unit is 4 ns. | 0x00000000 |

### 7.9.15.4 DEVCPU_GCB:PTP_TIMERS:PTP_TIMER_CTRL

**Parent:** DEVCPU_GCB:PTP_TIMERS

**Instances:** 1

Control register for PTP timers

*Table 468* • **Fields in PTP_TIMER_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_LATCH | 2 | One-shot | Latch time of day counter at the same time as the delay timer.<br><br>0: No action.<br>1: The time of day counter and the delay timer are latched at the same time. The results are stored in PTP_TOD_SECS, PTP_TOD_NANOSECS, and PTP_DELAY. | 0x0 |
| PTP_TIMER_ENA | 1 | R/W | Enable delay timer. | 0x0 |
| PTP_TOD_RST | 0 | One-shot | Reset the seconds fraction of the time of day counter. | 0x0 |

## 7.9.16 DEVCPU_GCB:MEMITGR

**Parent:** DEVCPU_GCB

**Instances:** 1

The memory integrity monitor is associated with one or more memories with build-in parity-protection and/or error-correction logic. Through the integrity monitor, address locations of failures and/or corrections can be read out.

There may be more than one integrity controller in the design, also - not all memories has an associated controller.

*Table 469* • **Registers in MEMITGR**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MEMITGR_CTRL | 0x00000000 | 1 | Monitor control | Page 377 |
| MEMITGR_STAT | 0x00000004 | 1 | Monitor status | Page 378 |
| MEMITGR_INFO | 0x00000008 | 1 | Memory indication | Page 379 |
| MEMITGR_IDX | 0x0000000C | 1 | Memory index | Page 380 |

### 7.9.16.1 DEVCPU_GCB:MEMITGR:MEMITGR_CTRL

**Parent:** DEVCPU_GCB:MEMITGR

**Instances:** 1

*Table 470 •* **Fields in MEMITGR_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTIVATE | 0 | One-shot | Setting this field transitions the integrity monitor between operating modes. Transitioning between modes takes time, this field remains set until the new mode is reached. During this time the monitor also reports busy (MEMITGR_MODE.MODE_BUSY is set).<br>From IDLE (MEMITGR_MODE.MODE_IDLE is set) the monitor can transition into either DETECT or LISTEN mode, the DETECT mode is entered if a memory reports an indication - the LISTEN mode is entered if no indications are reported. The first time after reset the monitor will not detect indications, that is; it will transition directly from IDLE to LISTEN mode.<br>From DETECT (MEMITGR_MODE.MODE_DETECT is set) the monitor can transition into either DETECT or LISTEN mode, the DETECT mode is entered if more indications are reported - the LISTEN mode is entered if no more indications are reported.<br>From LISTEN (MEMITGR_MODE.MODE_LISTEN is set) the monitor can transition into IDLE mode. | 0x0 |

### 7.9.16.2 DEVCPU_GCB:MEMITGR:MEMITGR_STAT

**Parent:** DEVCPU_GCB:MEMITGR

**Instances:** 1

*Table 471 •* **Fields in MEMITGR_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INDICATION | 4 | R/O | If this field is set then there is an indication from one of the memories that needs to be analyzed. An indication is either a parity detection or an error correction.<br>This field is only set when the monitor is in LISTEN mode (MEMITGR_MODE.MODE_LISTEN is set), in all other states (including BUSY) this field returns 0. | 0x0 |
| MODE_LISTEN | 3 | R/O | This field is set when the monitor is in LISTEN mode, during listen mode the monitor continually check for parity/correction indications from the memories. | 0x0 |
| MODE_DETECT | 2 | R/O | This field is set when the monitor is in DETECT mode, during detect mode the MEMITGR_INFO register contains valid information about one indication. | 0x0 |
| MODE_IDLE | 1 | R/O | This field is set when the monitor is in IDLE mode. | 0x1 |
| MODE_BUSY | 0 | R/O | The busy signal is a copy of the MEMITGR_CTRL.ACTIVATE field, see description of that field for more information about the different states/modes of the monitor. | 0x0 |

### 7.9.16.3 DEVCPU_GCB:MEMITGR:MEMITGR_INFO

**Parent:** DEVCPU_GCB:MEMITGR

**Instances:** 1

This field is only valid when the monitor is in the DETECT (MEMITGR_MODE.MODE_DETECT is set) mode.

*Table 472 •* **Fields in MEMITGR_INFO**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MEM_ERR | 31 | R/O | This field is set if the monitor has detected a parity indication (or an unrecoverable correction). | 0x0 |
| MEM_COR | 30 | R/O | This field is set if the monitor has detected a correction. | 0x0 |

*Table 472 •* **Fields in MEMITGR_INFO *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MEM_ERR_OVF | 29 | R/O | This field is set if the monitor has detected a parity indication (or an unrecoverable correction) for which the address has not been recorded.<br>If MEMITGR_INFO.MEM_ERR is set then there has been more than one indication, then only the address of the newest indication has been kept.<br>If MEMITGR_INFO.MEM_ERR is cleared then an indication has occurred for which the address could not be stored, this is a very rare situation that can only happen if an indication is detected just as the memory is talking to the monitor. | 0x0 |
| MEM_COR_OVF | 28 | R/O | This field is set if the monitor has correction indication for which the address has not been recorded.<br>If MEMITGR_INFO.MEM_ERR is set then there has also been a parity indication (or an unrecoverable correction) which takes priority over correction indications.<br>If MEMITGR_INFO.MEM_ERR is cleared and MEMITGR_INFO.MEM_COR is set then there has been more than one correction indication, then only the address of the newest correction indication has been kept.<br>If MEMITGR_INFO.MEM_ERR and MEMITGR_INFO.MEM_COR is both cleared then a correction indication has occurred for which the address could not be stored, this is a very rare situation that can only happen if an indication is detected just as the memory is talking to the monitor. | 0x0 |
| MEM_ADDR | 27:0 | R/O | This field is valid only when MEMITGR.MEM_ERR or MEMITGR.MEM_COR is set. | 0x0000000 |

### 7.9.16.4 DEVCPU_GCB:MEMITGR:MEMITGR_IDX

**Parent:** DEVCPU_GCB:MEMITGR

**Instances:** 1

This field is only valid when the monitor is in the DETECT (MEMITGR_MODE.MODE_DETECT is set) mode.

*Table 473 •* **Fields in MEMITGR_IDX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MEM_IDX | 15:0 | R/O | This field contains a unique index for the memory for which info is currently provided in MEMITGR_MEMINFO. Indexes are counted from 1 (not 0). | 0x0000 |

# 7.10 DEVCPU_QS

*Table 474 •* **Register Groups in DEVCPU_QS**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| XTR | 0x00000000 | 1 | Frame Extraction Related Registers | Page 381 |
| INJ | 0x00000034 | 1 | Frame Injection Related Registers | Page 384 |

## 7.10.1 DEVCPU_QS:XTR

**Parent:** DEVCPU_QS

**Instances:** 1

CPU queue system registers related to frame extraction.

*Table 475 •* **Registers in XTR**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| XTR_FRM_PRUNING | 0x00000000 | 2<br>0x00000004 | Frame Pruning | Page 381 |
| XTR_GRP_CFG | 0x00000008 | 2<br>0x00000004 | Group Configuration | Page 382 |
| XTR_MAP | 0x00000010 | 2<br>0x00000004 | Map Queue to Group | Page 382 |
| XTR_RD | 0x00000018 | 2<br>0x00000004 | Read from Group FIFO | Page 383 |
| XTR_QU_FLUSH | 0x00000028 | 1 | Queue Flush | Page 383 |
| XTR_DATA_PRESENT | 0x0000002C | 1 | Extraction Status | Page 384 |

### 7.10.1.1 DEVCPU_QS:XTR:XTR_FRM_PRUNING

**Parent:** DEVCPU_QS:XTR

**Instances:** 2

*Table 476 •* **Fields in XTR_FRM_PRUNING**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PRUNE_SIZE | 7:0 | R/W | Extracted frames for the corresponding queue are pruned PRUNE_SIZE 32-bit words.<br><br>Note : PRUNE_SIZE is the frame data size, including the IFH.<br>0 : No pruning<br>1: Frames extracted are pruned to 8 bytes.<br>2: Frames extracted are pruned to 12 bytes.<br>.<br>'0xFF': Frames extracted are pruned to 1024 bytes | 0x00 |

### 7.10.1.2 DEVCPU_QS:XTR:XTR_GRP_CFG

**Parent:** DEVCPU_QS:XTR

**Instances:** 2

*Table 477 •* **Fields in XTR_GRP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BYTE_SWAP | 0 | R/W | Controls - per extraction group - the byte order of the data word read in XTR_RD. When using little-Endian mode, then the first byte of the destination MAC address is placed at XTR_RD[7:0]. When using network-order, then the first byte of the destination MAC address is placed at XTR_RD[31:25].<br>0: Network-order (big-endian).<br>1: Little-endian. | 0x1 |
| STATUS_WORD_POS | 1 | R/W | Select order of last data and status words.<br>0: Status just before last data.<br>1: Status just after last data. | 0x1 |

### 7.10.1.3 DEVCPU_QS:XTR:XTR_MAP

**Parent:** DEVCPU_QS:XTR

**Instances:** 2

*Table 478* • **Fields in XTR_MAP**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GRP | 4 | R/W | Maps a queue to a certain extractor group | 0x0 |
| MAP_ENA | 0 | R/W | Enables extraction of a queue.<br><br>Disabling of extraction for a queue happens upon next frame boundary. That is, a frame being extracted at the time of queue disabling is not affected.<br>'0' : Queue is not mapped to a queue group ( queue is disabled )<br>'1' : Queue is mapped to the queue group defined by XTR::XTR_MAP ( queue is enabled ) | 0x0 |

### 7.10.1.4 DEVCPU_QS:XTR:XTR_RD

**Parent:** DEVCPU_QS:XTR

**Instances:** 2

*Table 479* • **Fields in XTR_RD**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DATA | 31:0 | R/O | Frame Data. Read from this register to obtain the next 32 bits of the frame data currently stored in the CPU queue system. Each read must check for the special values "0x8000000n", 0<=n<=7, as seen below;<br>Note that when a status word is presented, it can be put just before or just after the last data (XTR_GRP_CFG).<br>n=0-3: EOF. Unused bytes in last is 'n'.<br>n=4 : EOF, but truncated.<br>n=5 : EOF Aborted. Frame invalid.<br>n=6 : Escape. Next read is packet data.<br>n=7 : Data not ready for reading out. | 0x00000000 |

### 7.10.1.5 DEVCPU_QS:XTR:XTR_QU_FLUSH

**Parent:** DEVCPU_QS:XTR

**Instances:** 1

*Table 480 •* **Fields in XTR_QU_FLUSH**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FLUSH | 1:0 | R/W | Enable software flushing of a CPU queue.<br>Note that before flushing the a CPU queue it may be necessary to stop the OQS from sending data into the CPU queues.<br><br>'0': No action<br>'1': Do CPU queue flushing | 0x0 |

### 7.10.1.6 DEVCPU_QS:XTR:XTR_DATA_PRESENT

**Parent:** DEVCPU_QS:XTR

**Instances:** 1

*Table 481 •* **Fields in XTR_DATA_PRESENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DATA_PRESENT | 3:2 | R/O | When a frame, which should be forwarded to software has been received by the CPU queue system, the corresponding bit is set. When software has extracted all frames from a CPU queue the bit is cleared, i.e. the bit remains set as long as at least one byte of frame data for the corresponding queue is present in the queue system.<br><br>Note : If a queue isn't map to a group DATA_PRESENT will be '0'<br>'0': No data available for this CPU queue<br>'1': At least one frame is available for this cpu queue | 0x0 |
| DATA_PRESENT_GRP | 1:0 | R/O | When a queue group has a frame present, the bit corresponding to the queue group number gets set. It remains set until all frame data have been extracted.<br>'0': No frames available for this CPU queue group.<br>'1': At least one frame is available for this CPU queue group. | 0x0 |

## 7.10.2 DEVCPU_QS:INJ

**Parent:** DEVCPU_QS

**Instances:** 1

CPU queue system registers related to frame injection.

*Table 482 •* **Registers in INJ**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| INJ_GRP_CFG | 0x00000000 | 2<br>0x00000004 | Group Configuration | Page 385 |
| INJ_WR | 0x00000008 | 2<br>0x00000004 | Write to Group FIFO | Page 385 |
| INJ_CTRL | 0x00000010 | 2<br>0x00000004 | Injection Control | Page 385 |
| INJ_STATUS | 0x00000018 | 1 | Injection Status | Page 386 |
| INJ_ERR | 0x0000001C | 2<br>0x00000004 | Injection Errors | Page 387 |

### 7.10.2.1 DEVCPU_QS:INJ:INJ_GRP_CFG

**Parent:** DEVCPU_QS:INJ

**Instances:** 2

*Table 483 •* **Fields in INJ_GRP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BYTE_SWAP | 8 | R/W | Controls - per injection group - the byte order of the data word in INJ_WR.<br>0: Network-order (big-endian).<br>1: Little-endian. | 0x1 |

### 7.10.2.2 DEVCPU_QS:INJ:INJ_WR

**Parent:** DEVCPU_QS:INJ

**Instances:** 2

*Table 484 •* **Fields in INJ_WR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DATA | 31:0 | R/W | Frame Write.Write to this register inject the next 32 bits of the frame data currently injected into the chip. | 0x00000000 |

### 7.10.2.3 DEVCPU_QS:INJ:INJ_CTRL

**Parent:** DEVCPU_QS:INJ

**Instances:** 2

*Table 485 •* **Fields in INJ_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GAP_SIZE | 28:21 | R/W | It is allowed to inject a number of "dummy" bytes in front of a frame before the actual frame data. The number of bytes that should be discarded is specified with this field. | 0x00 |
| ABORT | 20 | One-shot | Abort frame currently injected.<br>Write:<br>'0': No action<br>'1': Frame currently injected is aborted (Bit is automatically cleared) | 0x0 |
| EOF | 19 | One-shot | EOF must be set before last data of a frame is injected.<br>'0': No action<br>'1': Next word is the last word of the frame injected | 0x0 |
| SOF | 18 | One-shot | SOF must be set before injecting a frame.<br>Write:<br>'0': No action<br>'1': Start of new frame injection<br><br>Read:<br>'0': First data word has been moved to the IQS.<br>'1': First data word has not been moved to the IQS. | 0x0 |
| VLD_BYTES | 17:16 | R/W | The number of valid bytes in the last word must be set before last data of a frame is injected.<br>0: Bits 31-0 in the last word are valid.<br>1: Bits 31-24 in the last word are valid.<br>2: Bits 31-16 in the last word are valid.<br>3: Bits 31-7 in the last word are valid.<br>This encoding applies when big-endian is used for INJ_WR. | 0x0 |

### 7.10.2.4 DEVCPU_QS:INJ:INJ_STATUS

**Parent:** DEVCPU_QS:INJ

**Instances:** 1

*Table 486 •* **Fields in INJ_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WMARK_REACHED | 5:4 | R/O | Before the CPU injects a frame, software may check if the input queue has reached high watermark. If the watermark in the IQS has been reached this bit will be set.<br>'0': Input queue has not reached high watermark<br>'1': Input queue has reached high watermark, and frames injected may be dropped due to buffer overflow. | 0x0 |
| FIFO_RDY | 3:2 | R/O | When '1' the injector group's FIFO is ready for additional data written through the INJ_WR register.<br>'0': The injector group cannot accept additional data.<br>'1': The injector group is able to accept additional data. | 0x0 |
| INJ_IN_PROGRESS | 1:0 | R/O | When '1' the injector group is in the process of receiving a frame, and at least one write to INJ_WR remains before the frame is forwarded to the front ports. When '0' the injector group is waiting for an initiation of a frame injection.<br>'0': A frame injection is not in progress.<br>'1': A frame injection is in progress. | 0x0 |

### 7.10.2.5 DEVCPU_QS:INJ:INJ_ERR

**Parent:** DEVCPU_QS:INJ

**Instances:** 2

The bits in this register are cleared by writing a '1' to the relevant bit-positions.

*Table 487 •* **Fields in INJ_ERR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ABORT_ERR_STICKY | 1 | Sticky | If the CPU aborts an on-going frame injection by a '1' to INJ_CTRL::ABORT, the on-going frame injection is aborted and the injection controller prepares for a new injection. This situation could indicate a software error.<br>'0': No error.<br>'1': Previous frame was aborted with a write to INJ_CTRL::ABORT or due to an internal error. | 0x0 |
| WR_ERR_STICKY | 0 | Sticky | If the CPU writes to INJ_WR without having initiated a frame injection with INJ_CTRL, this sticky bit gets set.<br>'0': No error.<br>'1': Erroneous write to INJ_WR has been made. | 0x0 |

## 7.11 DEVCPU_PI

*Table 488 •* **Register Groups in DEVCPU_PI**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PI | 0x00000000 | 1 | Registers for the parallel interface | Page 388 |

### 7.11.1 DEVCPU_PI:PI

**Parent:** DEVCPU_PI

**Instances:** 1

Registers for the parallel interface. These registers are only reachable via the parallel interface. None of the settings in these register applies to anything else than the parallel interface when it operates in slave mode.

*Table 489 •* **Registers in PI**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PI_CTRL | 0x00000000 | 1 | Control of PI accesses | Page 389 |
| PI_CFG | 0x00000004 | 1 | Configuration of PI accesses | Page 390 |
| PI_STAT | 0x00000008 | 1 | Status for PI accesses | Page 391 |

*Table 489 •* **Registers in PI *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PI_MODE | 0x0000000C | 1 | Mode of the parallel interface | Page 391 |
| PI_SLOW_DATA | 0x00000010 | 2<br>0x00000004 | Slow Data | Page 392 |

### 7.11.1.1 DEVCPU_PI:PI:PI_CTRL

**Parent:** DEVCPU_PI:PI

**Instances:** 1

*Table 490 •* **Fields in PI_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SLOW_IDX | 1 | R/W | Use this field to select a destination index register for slow access results. By using different indexes it is possible to have more than one outstanding slow-access at any given time. This may be utilized by interrupt routines, just remember that an interrupt routine should restore this register to its previous value before exiting the routine.<br>Note: If multiple levels of interrupts is required, more than there are slow-access-indexes, then it is possible for the high-priority interrupt routine to use normal-accesses (by disabling slow-access via SLOW_ENA), then the PI will be occupied while reading - but that access will not interfere with any ongoing slow accesses. | 0x0 |

*Table 490 •* **Fields in PI_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SLOW_ENA | 0 | R/W | Set this field to enable slow accesses. For a normal accesses ("slow" is not enabled) the PI access will be stalled until data is ready to be read out of the device. When slow-data is enabled then a read from any register (except these PI registers) will return immediately - the read will then be processed will the external CPU is free to do something else. The field SLOW_IN_PROGRESS indicates when slow accesses are done, once the access has completed the result can be read from the SLOWDATA register at the index corresponding to the SLOW_IDX that was used when the access was initiated. When slow access is enabled, the data which is returned when the access is started is actually the result from the corresponding SLOWDATA register, this means that it is possible to do "back-to-back" slow accesses, every time a new slow-access is started - the result of the old access is read out. | 0x0 |

### 7.11.1.2 DEVCPU_PI:PI:PI_CFG

**Parent:** DEVCPU_PI:PI

**Instances:** 1

*Table 491 •* **Fields in PI_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BUSY_FEEDBACK_ENA | 5 | R/W | Set this field to enable busy feedback to the physical PI. When set origin-busy causes the physical interface to delay sampling of data (and generating of ndone). | 0x1 |
| WR_ACK_ENA | 4 | R/W | Set this field to hold write accesses until the write-request has reached the target. By default write accesses is completed as soon as the write is detected (by the PI). | 0x0 |

*Table 491 •* **Fields in PI_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PI_WAIT | 3:0 | R/W | Configures the delay from detecting asserted PI_nCS until the chip samples the control signals. The delay is configured in steps of 8ns. This field should be lowered to match the performance and interface timing of the external CPU.<br>This field can be set to zero, in that case the control signals will be sampled immediately when asserted PI_nCS is detected. | 0xD |

### 7.11.1.3 DEVCPU_PI:PI:PI_STAT

**Parent:** DEVCPU_PI:PI

**Instances:** 1

*Table 492 •* **Fields in PI_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ORIGIN_ERR_STICKY | 6 | Sticky | This field is set when accessing an unknown target or an unknown address inside a known target. | 0x0 |
| SLOW_BUSY_STICKY | 5:4 | Sticky | This field is set if a new access has been started on a busy slow index (each bit in this field correspond to a slow index). | 0x0 |
| SLOW_BUSY | 3:2 | R/O | This field indicates if a slow access is in progress. When a bit is set in this field, the corresponding slow access index is currently occupied by an access. | 0x0 |
| SLOW_DONE | 1:0 | R/O | This field indicates if slow-data is pending: When a bit is set in this field, the corresponding slow access index contains unread data. The bits in this field is cleared when the corresponding slow-data index is read. | 0x0 |

### 7.11.1.4 DEVCPU_PI:PI:PI_MODE

**Parent:** DEVCPU_PI:PI

**Instances:** 1

In order for the configuration to work independently of the current transfer mode; The 8 low bits of this register must be mirrored throughout the entire 32-bit dataword when writing. Also the configuration must be written twice, this ensures that an 8-bit interface correctly receives configuration from a 16-bit external CPU.

For example: For default nDone polarity, big-endian mode, auto-address mode, and 16-bit data bus the low 8-bit of this register will be 0x0A. Then the actual 32-bit write value is 0x0A0A0A0A.

*Table 493 •* **Fields in PI_MODE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DATA_BUS_WID | 3 | R/W | This field configures the data-width of the PI interface. Either 8-bit or 16-bit data-bus is supported. By default the width is 8-bit, thus a 16-bit processor has to configure this field to use the entire bus width.<br>0 : Data bus is 8 bit wide<br>1 : Data bus is 16 bit wide | 0x0 |
| ADDR_AUTO_DIS | 2 | R/W | Disables automatic tracking of sub-word addresses. By default the low two address bits are not needed, the device keeps track of addresses inside 32-bit words and aligns data accordingly. | 0x0 |
| ENDIAN | 1 | R/W | Configure the byte order mode on the parallel interface.<br>0 : Little Endian<br>1 : Big Endian | 0x0 |
| NDONE_POL | 0 | R/W | Configures the nDone pin's active level.<br>0 : nDone pin is active when low<br>1 : nDone pin is active when high | 0x0 |

#### 7.11.1.5 DEVCPU_PI:PI:PI_SLOW_DATA

**Parent:** DEVCPU_PI:PI

**Instances:** 2

*Table 494 •* **Fields in PI_SLOW_DATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PI_SLOW_DATA | 31:0 | R/W | When a slow access is done, the result is stored in this register. | 0x00000000 |

## 7.12 HSIO

Register Collection for Control of Macros (SERDES1G, SERDES6G, LCPLL)

*Table 495 •* **Register Groups in HSIO**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PLL5G_CFG | 0x00000000 | 1 | PLL5G Configuration Registers | Page 393 |
| PLL5G_STATUS | 0x00000018 | 1 | PLL5G Status Registers | Page 394 |
| RCOMP_STATUS | 0x00000024 | 1 | RCOMP Status Registers | Page 395 |

*Table 495 •* **Register Groups in HSIO *(continued)***

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES1G_ANA_CFG | 0x0000002C | 1 | SERDES1G Analog Configuration Registers | Page 395 |
| SERDES1G_DIG_CFG | 0x00000048 | 1 | SERDES1G Digital Configuration Register | Page 401 |
| SERDES1G_DIG_STATUS | 0x0000005C | 1 | SERDES1G Digital Status Register | Page 402 |
| MCB_SERDES1G_CFG | 0x00000060 | 1 | MCB SERDES1G Configuration Register | Page 402 |
| SERDES6G_ANA_CFG | 0x00000064 | 1 | SERDES6G Analog Configuration Registers | Page 403 |
| SERDES6G_DIG_CFG | 0x00000088 | 1 | SERDES6G Digital Configuration Registers | Page 409 |
| MCB_SERDES6G_CFG | 0x000000AC | 1 | MCB SERDES6G Configuration Register | Page 411 |

## 7.12.1 HSIO:PLL5G_CFG

**Parent:** HSIO

**Instances:** 1

Configuration register set for PLL5G.

*Table 496 •* **Registers in PLL5G_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PLL5G_CFG0 | 0x00000000 | 1 | PLL5G Configuration 0 | Page 393 |

### 7.12.1.1 HSIO:PLL5G_CFG:PLL5G_CFG0

**Parent:** HSIO:PLL5G_CFG

**Instances:** 1

Configuration register 0 for PLL5G

*Table 497 •* **Fields in PLL5G_CFG0**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 5:0 | R/W | Must be set to its default. | 0x05 |
| CPU_CLK_DIV | 11:6 | R/W | Setting for CPU clock divider<br>5: 250 MHz<br>6: 416.66 MHz<br>14: 312.50 MHz<br>Others: Reserved | 0x05 |
| RESERVED | 12 | R/W | Must be set to its default. | 0x1 |

*Table 497 •* **Fields in PLL5G_CFG0 *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 13 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 14 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 15 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 17:16 | R/W | Must be set to its default. | 0x2 |
| RESERVED | 22:18 | R/W | Must be set to its default. | 0x0D |
| RESERVED | 26:23 | R/W | Must be set to its default. | 0x7 |
| RESERVED | 28 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 29 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 30 | R/W | Must be set to its default. | 0x1 |

## 7.12.2 HSIO:PLL5G_STATUS

**Parent:** HSIO

**Instances:** 1

Status register set for PLL5G.

*Table 498 •* **Registers in PLL5G_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PLL5G_STATUS0 | 0x00000000 | 1 | PLL5G Status 0 | Page 394 |

### 7.12.2.1 HSIO:PLL5G_STATUS:PLL5G_STATUS0

**Parent:** HSIO:PLL5G_STATUS

**Instances:** 1

Status register 0 for the PLL5G

*Table 499 •* **Fields in PLL5G_STATUS0**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LOCK_STATUS | 0 | R/O | PLL lock status 0: not locked, 1: locked | 0x0 |
| READBACK_DATA | 8:1 | R/O | RCPLL Interface to read back internal data of the FSM. | 0x00 |
| CALIBRATION_DONE | 9 | R/O | RCPLL Flag that indicates that the calibration procedure has finished. | 0x0 |
| CALIBRATION_ERR | 10 | R/O | RCPLL Flag that indicates errors that may occur during the calibration procedure. | 0x0 |
| OUT_OF_RANGE_ERR | 11 | R/O | RCPLL Flag that indicates a out of range condition while NOT in calibration mode. | 0x0 |

*Table 499 •* **Fields in PLL5G_STATUS0 *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RANGE_LIM | 12 | R/O | RCPLL Flag range limiter signaling | 0x0 |

## 7.12.3 HSIO:RCOMP_STATUS

**Parent:** HSIO

**Instances:** 1

Status register set for RCOMP.

*Table 500 •* **Registers in RCOMP_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| RCOMP_STATUS | 0x00000000 | 1 | RCOMP Status | Page 395 |

### 7.12.3.1 HSIO:RCOMP_STATUS:RCOMP_STATUS

**Parent:** HSIO:RCOMP_STATUS

**Instances:** 1

Status register bits for the RCOMP

*Table 501 •* **Fields in RCOMP_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BUSY | 12 | R/O | Resistor comparison activity<br>0: resistor measurement finished or inactive<br>1: resistor measurement in progress | 0x0 |
| DELTA_ALERT | 7 | R/O | Alarm signal if rcomp isn't best choice anymore<br>0: inactive<br>1: active | 0x0 |
| RCOMP | 3:0 | R/O | Measured resistor value<br>0: maximum resistance value<br>15: minimum resistance value | 0x0 |

## 7.12.4 HSIO:SERDES1G_ANA_CFG

**Parent:** HSIO

**Instances:** 1

Configuration register set for SERDES1G (analog parts)

*Table 502* • **Registers in SERDES1G_ANA_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES1G_DES_CFG | 0x00000000 | 1 | SERDES1G Deserializer Cfg | Page 396 |
| SERDES1G_IB_CFG | 0x00000004 | 1 | SERDES1G Input Buffer Cfg | Page 397 |
| SERDES1G_OB_CFG | 0x00000008 | 1 | SERDES1G Output Buffer Cfg | Page 398 |
| SERDES1G_SER_CFG | 0x0000000C | 1 | SERDES1G Serializer Cfg | Page 399 |
| SERDES1G_COMMON_CFG | 0x00000010 | 1 | SERDES1G Common Cfg | Page 400 |
| SERDES1G_PLL_CFG | 0x00000014 | 1 | SERDES1G Pll Cfg | Page 400 |

### 7.12.4.1 HSIO:SERDES1G_ANA_CFG:SERDES1G_DES_CFG

**Parent:** HSIO:SERDES1G_ANA_CFG

**Instances:** 1

Configuration register for SERDES1G deserializer

*Table 503* • **Fields in SERDES1G_DES_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DES_PHS_CTRL | 16:13 | R/W | Control of phase regulator logic.<br>Bit 3 must always be set to 0.<br>Optimal settings for bits 2:0 are 4 through 7; recommended setting is 6.<br>0: Disabled<br>1: Enabled with 99 ppm limit<br>2: Enabled with 202 ppm limit<br>3: Enabled with 485 ppm limit<br>4: Enabled if corresponding PCS is in sync with 50 ppm limit<br>5: Enabled if corresponding PCS is in sync with 99 ppm limit<br>6: Enabled if corresponding PCS is in sync with 202 ppm limit<br>7: Enabled if corresponding PCS is in sync with 485 ppm limit | 0x0 |
| RESERVED | 12:11 | R/W | Must always be set to its default. | 0x0 |

*Table 503 •* **Fields in SERDES1G_DES_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DES_MBTR_CTRL | 10:8 | R/W | Des phase control for 180 degrees deadlock block mode of operation<br>000: Depending on density of input pattern<br>001: Active until PCS has synchronized<br>010: Depending on density of input pattern until PCS has synchronized<br>011: Never<br>100: Always<br>All other settings are reserved. | 0x0 |
| DES_BW_ANA | 7:5 | R/W | Bandwidth selection for proportional path of the CDR loop.<br>0: Reserved<br>1: Reserved<br>2: Reserved<br>3: Reserved<br>4: Divide by 16<br>5: Divide by 32<br>6: Divide by 64<br>7: Divide by 128 | 0x0 |
| RESERVED | 4 | R/W | Must be set to its default. | 0x0 |
| DES_BW_HYST | 3:1 | R/W | Selection of time constant for integrative path of CDR loop.<br>0: Reserved<br>1: Reserved<br>2: Reserved<br>3: Divide by 16<br>4: Divide by 32<br>5: Divide by 64<br>6: Divide by 128<br>7: Divide by 256 | 0x0 |

### 7.12.4.2 HSIO:SERDES1G_ANA_CFG:SERDES1G_IB_CFG

**Parent:** HSIO:SERDES1G_ANA_CFG

**Instances:** 1

Configuration register for SERDES1G input buffer

*Table 504 •* **Fields in SERDES1G_IB_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IB_FX100_ENA | 27 | R/W | Switches signal detect circuit into low frequency mode, must be used in fx100 mode | 0x0 |
| IB_DET_LEV | 20:19 | R/W | Detect thresholds.<br>00: 159-189mVppd<br>01: 138-164mVppd<br>10: 109-124mVppd<br>11: 74-89mVppd | 0x0 |
| IB_HYST_LEV | 14 | R/W | Input buffer hysteresis levels.<br>0: 59-79mV<br>1: 81-124mV | 0x0 |

*Table 504 •* **Fields in SERDES1G_IB_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IB_ENA_CMV_TERM | 13 | R/W | Enable common mode voltage termination<br>0: Low termination ($V_{DD_A}$ × 0.7)<br>1: High termination ($V_{DD_A}$) | 0x0 |
| IB_ENA_DC_COUPLING | 12 | R/W | Enable dc-coupling of input signal<br>0: Disable<br>1: Enable | 0x0 |
| IB_ENA_DETLEV | 11 | R/W | Enable detect level circuit<br>0: Disable<br>1: Enable | 0x0 |
| IB_ENA_HYST | 10 | R/W | Enable hysteresis for input signal. Hysteresis can only be enabled if DC offset compensation (bit 9) is disabled.<br>0: Disable<br>1: Enable | 0x0 |
| IB_ENA_OFFSET_COMP | 9 | R/W | Enable offset compensation of input stage. This bit must be disabled to enable hysteresis (bit 10).<br>0: Disable<br>1: Enable | 0x0 |
| IB_EQ_GAIN | 8:6 | R/W | Selects weighting between AC and DC input path.<br>0: Reserved<br>1: Reserved<br>2: 0dB (recommended value)<br>3: 1.5dB<br>4: 3dB<br>5: 6dB<br>6: 9dB<br>7: 12.5dB | 0x0 |
| IB_SEL_CORNER_FREQ | 5:4 | R/W | Corner frequencies of AC path.<br>0: 1.3GHz<br>1: 1.5GHz<br>2: 1.6GHz<br>3: 1.8GHz | 0x0 |
| IB_RESISTOR_CTRL | 3:0 | R/W | Resistor control. Value must be taken from RCOMP_STATUS.RCOMP. | 0x0 |

### 7.12.4.3 HSIO:SERDES1G_ANA_CFG:SERDES1G_OB_CFG

**Parent:** HSIO:SERDES1G_ANA_CFG

**Instances:** 1

Configuration register for SERDES1G output buffer

*Table 505 •* **Fields in SERDES1G_OB_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OB_SLP | 18:17 | R/W | Slope/slew rate control.<br>0: 45ps<br>1: 85ps<br>2: 105ps<br>3: 115ps | 0x0 |
| OB_AMP_CTRL | 16:13 | R/W | Amplitude control, in steps of 50mVppd.<br>0: 0.4Vppd<br>15: 1.1Vppd | 0x0 |
| RESERVED | 12:10 | R/W | Must be set to its default. | 0x2 |
| RESERVED | 9:8 | R/W | Must be set to its default. | 0x0 |
| OB_VCM_CTRL | 7:4 | R/W | Common mode voltage control.<br>0: Reserved<br>1: 440mV<br>2: 480mV<br>3: 460mV<br>4: 530mV<br>5: 500mV<br>6: 570mV<br>7: 550mV | 0x4 |
| OB_RESISTOR_CTRL | 3:0 | R/W | Resistor control. Value must be taken from RCOMP_STATUS.RCOMP. | 0x0 |

### 7.12.4.4 HSIO:SERDES1G_ANA_CFG:SERDES1G_SER_CFG

**Parent:** HSIO:SERDES1G_ANA_CFG

**Instances:** 1

Configuration register for SERDES1G serializer

*Table 506 •* **Fields in SERDES1G_SER_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SER_IDLE | 9 | R/W | Invert output D0b for idle-mode of OB<br>0: Non-inverting<br>1. Inverting | 0x0 |
| SER_DEEMPH | 8 | R/W | Invert and delays (one clk cycle) output D1 for de-emphasis of OB<br>0: Non-inverting and non-delaying<br>1: Inverting and delaying | 0x0 |
| RESERVED | 7:4 | R/W | Must be set to its default. | 0x0 |
| SER_ENHYS | 3 | R/W | Enable hysteresis for phase alignment<br>0: Disable hysteresis<br>1: Enable hysteresis | 0x0 |

*Table 506 •* **Fields in SERDES1G_SER_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SER_BIG_WIN | 2 | R/W | Use wider window for phase alignment<br>0: Use small window for low jitter (100 to 200ps)<br>1: Use wide window for higher jitter (150 to 300 ps) | 0x0 |
| SER_EN_WIN | 1 | R/W | Enable window for phase alignment<br>0: Disable window<br>1: Enable window | 0x0 |
| SER_ENALI | 0 | R/W | Enable phase alignment<br>0: Disable phase alignment<br>1: Enable phase alignment | 0x0 |

### 7.12.4.5 HSIO:SERDES1G_ANA_CFG:SERDES1G_COMMON_CFG

**Parent:** HSIO:SERDES1G_ANA_CFG

**Instances:** 1

Configuration register for common SERDES1G functions Note: When enabling the facility loop (ena_floop) also the phase alignment in the serializer has to be enabled and configured adequate.

*Table 507 •* **Fields in SERDES1G_COMMON_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SYS_RST | 31 | R/W | System reset (low active)<br>0: Apply reset (not self-clearing)<br>1: Reset released | 0x0 |
| ENA_LANE | 18 | R/W | Enable lane<br>0: Disable lane<br>1: Enable line | 0x0 |
| RESERVED | 17:12 | R/W | Must be set to its default. | 0x00 |
| ENA_ELOOP | 11 | R/W | Enable equipment loop<br>0: Disable<br>1: Enable | 0x0 |
| ENA_FLOOP | 10 | R/W | Enable facility loop<br>0: Disable<br>1: Enable | 0x0 |
| RESERVED | 9:8 | R/W | Must be set to its default. | 0x0 |
| RESERVED | 7 | R/W | Must be set to its default. | 0x1 |
| RESERVED | 0 | R/W | Must be set to its default. | 0x1 |

### 7.12.4.6 HSIO:SERDES1G_ANA_CFG:SERDES1G_PLL_CFG

**Parent:** HSIO:SERDES1G_ANA_CFG

**Instances:** 1

Configuration register for SERDES1G RCPLL

*Table 508 •* **Fields in SERDES1G_PLL_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 22:21 | R/W | Must be set to its default. | 0x0 |
| PLL_FSM_CTRL_DATA | 15:8 | R/W | Control data for FSM | 0x00 |
| PLL_FSM_ENA | 7 | R/W | Enable FSM | 0x0 |
| RESERVED | 6:5 | R/W | Must be set to its default. | 0x0 |
| RESERVED | 3 | R/W | Must be set to its default. | 0x0 |

## 7.12.5 HSIO:SERDES1G_DIG_CFG

**Parent:** HSIO

**Instances:** 1

Configuration register set for SERDES1G digital BIST and DFT functions.

*Table 509 •* **Registers in SERDES1G_DIG_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES1G_MISC_CFG | 0x00000010 | 1 | SERDES1G Misc Configuration | Page 401 |

### 7.12.5.1 HSIO:SERDES1G_DIG_CFG:SERDES1G_MISC_CFG

**Parent:** HSIO:SERDES1G_DIG_CFG

**Instances:** 1

Configuration register for miscellaneous functions

*Table 510 •* **Fields in SERDES1G_MISC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DES_100FX_CPMD_ENA | 8 | R/W | Enable deserializer cp/md handling for 100fx mode<br>0: Disable<br>1: Enable | 0x0 |
| RX_LPI_MODE_ENA | 5 | R/W | Enable RX-Low-Power feature (Power control by LPI-FSM in connected PCS)<br>0: Disable<br>1: Enable | 0x0 |
| TX_LPI_MODE_ENA | 4 | R/W | Enable TX-Low-Power feature (Power control by LPI-FSM in connected PCS)<br>0: Disable<br>1: Enable | 0x0 |

*Table 510 •* **Fields in SERDES1G_MISC_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_DATA_INV_ENA | 3 | R/W | Enable data inversion received from Deserializer<br>0: Disable<br>1: Enable | 0x0 |
| TX_DATA_INV_ENA | 2 | R/W | Enable data inversion sent to Serializer<br>0: Disable<br>1: Enable | 0x0 |
| LANE_RST | 0 | R/W | Lane Reset<br>0: No reset<br>1: Reset (not self-clearing) | 0x0 |

## 7.12.6 HSIO:SERDES1G_DIG_STATUS

**Parent:** HSIO

**Instances:** 1

Status register set for SERDES1G digital BIST and DFT functions.

*Table 511 •* **Registers in SERDES1G_DIG_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES1G_DFT_STATUS | 0x00000000 | 1 | SERDES1G DFT Status | Page 402 |

### 7.12.6.1 HSIO:SERDES1G_DIG_STATUS:SERDES1G_DFT_STATUS

**Parent:** HSIO:SERDES1G_DIG_STATUS

**Instances:** 1

Status register of SERDES1G DFT functions

*Table 512 •* **Fields in SERDES1G_DFT_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BIST_NOSYNC | 2 | R/O | BIST sync result<br>0: Synchronization successful<br>1: Synchronization on BIST data failed | 0x0 |

## 7.12.7 HSIO:MCB_SERDES1G_CFG

**Parent:** HSIO

**Instances:** 1

All SERDES1G macros are accessed via the serial Macro Configuration Bus (MCB). Each macro is configured by one MCB slave. All MCB slaves are connected in a daisy-chain loop.

*Table 513* • **Registers in MCB_SERDES1G_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MCB_SERDES1G_ADDR_CFG | 0x00000000 | 1 | MCB SERDES1G Address Cfg | Page 403 |

### 7.12.7.1 HSIO:MCB_SERDES1G_CFG:MCB_SERDES1G_ADDR_CFG

**Parent:** HSIO:MCB_SERDES1G_CFG

**Instances:** 1

Configuration of SERDES1G MCB slaves to be accessed

*Table 514* • **Fields in MCB_SERDES1G_ADDR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SERDES1G_WR_ONE_SHOT | 31 | One-shot | Initiate a write access to marked SERDES1G slaves<br>0: No write operation pending<br>1: Initiate write to slaves (kept 1 until write operation has finished) | 0x0 |
| SERDES1G_RD_ONE_SHOT | 30 | One-shot | Initiate a read access to marked SERDES1G slaves<br>0: No read operation pending (read op finished after bit has been set)<br>1: Initiate a read access (kept 1 until read operation has finished) | 0x0 |
| SERDES1G_ADDR | 24:0 | R/W | Activation vector for SERDES1G-Slaves, one-hot coded, each bit is related to one macro, e.g. bit 0 enables/disables access to macro No. 0<br>0: Disable macro access via MCB<br>1: Enable macro access via MCB | 0x1FFFFFF |

## 7.12.8 HSIO:SERDES6G_ANA_CFG

**Parent:** HSIO

**Instances:** 1

Configuration register set for SERDES6G (analog parts)

*Table 515* • **Registers in SERDES6G_ANA_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES6G_DES_CFG | 0x00000000 | 1 | SERDES6G Deserializer Cfg | Page 404 |

*Table 515 •* **Registers in SERDES6G_ANA_CFG *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES6G_IB_CFG | 0x00000004 | 1 | SERDES6G Input Buffer Cfg | Page 405 |
| SERDES6G_IB_CFG1 | 0x00000008 | 1 | SERDES6G Input Buffer Cfg1 | Page 406 |
| SERDES6G_OB_CFG | 0x0000000C | 1 | SERDES6G Output Buffer Cfg | Page 406 |
| SERDES6G_OB_CFG1 | 0x00000010 | 1 | SERDES6G Output Buffer Cfg1 | Page 407 |
| SERDES6G_SER_CFG | 0x00000014 | 1 | SERDES6G Serializer Cfg | Page 407 |
| SERDES6G_COMMON_CFG | 0x00000018 | 1 | SERDES6G Common Cfg | Page 408 |
| SERDES6G_PLL_CFG | 0x0000001C | 1 | SERDES6G Pll Cfg | Page 409 |

### 7.12.8.1 HSIO:SERDES6G_ANA_CFG:SERDES6G_DES_CFG

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register for SERDES6G deserializer

*Table 516 •* **Fields in SERDES6G_DES_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DES_PHS_CTRL | 16:13 | R/W | Control of phase regulator logic Bit 3 must always be set to 0.<br>Optimal settings for bits 2:0 are 4 through 7; recommended setting is 6.<br>0: Disabled<br>1: Enabled with 99 ppm limit<br>2: Enabled with 202 ppm limit<br>3: Enabled with 485 ppm limit<br>4: Enabled if corresponding PCS is in sync with 50 ppm limit<br>5: Enabled if corresponding PCS is in sync with 99 ppm limit<br>6: Enabled if corresponding PCS is in sync with 202 ppm limit<br>7: Enabled if corresponding PCS is in sync with 485 ppm limit | 0x0 |

*Table 516 •* **Fields in SERDES6G_DES_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DES_MBTR_CTRL | 12:10 | R/W | Des phase control for 180 degrees deadlock block mode of operation<br>000: Depending on density of input pattern<br>001: Active until PCS has synchronized<br>010: Depending on density of input pattern until PCS has synchronized<br>011: Never<br>100: Always<br>All other settings are reserved. | 0x0 |
| RESERVED | 9:8 | R/W | Must be set to its default. | 0x0 |
| DES_BW_HYST | 7:5 | R/W | Selection of time constant for integrative path of the CDR loop.<br>0: Reserved<br>1: Divide by 4<br>2: Divide by 8<br>3: Divide by 16<br>4: Divide by 32<br>5: Divide by 64<br>6: Divide by 128<br>7: Divide by 256<br>For more information about mode-dependent limitations, see SERDES6G Deserializer Configuration, page 28. | 0x0 |
| RESERVED | 4: | R/W | Must be set to its default. | 0x0 |
| DES_BW_ANA | 3:1 | R/W | Bandwidth selection for proportional path of CDR loop.<br>0: Reserved<br>1: Reserved<br>2: Divide by 4<br>3: Divide by 8<br>4: Divide by 16<br>5: Divide by 32<br>6: Divide by 64<br>7: Divide by 128<br>For more information about mode-dependent limitations, see SERDES6G Deserializer Configuration, page 28. | 0x0 |
| RESERVED | 0 | R/W | Must be set to its default. | 0x0 |

### 7.12.8.2 HSIO:SERDES6G_ANA_CFG:SERDES6G_IB_CFG

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register 0 for SERDES6G input buffer

*Table 517 •* **Fields in SERDES6G_IB_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 27:7 | R/W | Must be set to its default. | 0x00000 |
| IB_VBCOM | 6:4 | R/W | Level detection thresholds, in steps of approximately 8mV.<br>0: 60mV<br>7: 120mV | 0x0 |
| IB_RESISTOR_CTRL | 3:0 | R/W | Resistor control. Value must be taken from RCOMP_STATUS.RCOMP. | 0x0 |

### 7.12.8.3 HSIO:SERDES6G_ANA_CFG:SERDES6G_IB_CFG1

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register 1 for SERDES6G input buffer

*Table 518 •* **Fields in SERDES6G_IB_CFG1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 13:7 | R/W | Must be set to its default. | 0x00 |
| IB_CTERM_ENA | 5 | R/W | Common mode termination<br>0: Disable<br>1: Enable | 0x0 |
| IB_RESERVED | 4 | R/W | Must be set to 1. | 0x0 |
| IB_ENA_OFFSAC | 3 | R/W | Auto offset compensation for ac path<br>0: Disable<br>1: Enable | 0x0 |
| IB_ENA_OFFSDC | 2 | R/W | Auto offset compensation for dc path<br>0: Disable<br>1: Enable | 0x0 |
| IB_FX100_ENA | 1 | R/W | Increases timing constant for level detect circuit, must be used in FX100 mode<br>0: Normal speed<br>1: Slow speed (oversampling) | 0x0 |
| RESERVED | 0 | R/W | Must be set to its default. | 0x0 |

### 7.12.8.4 HSIO:SERDES6G_ANA_CFG:SERDES6G_OB_CFG

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register 0 for SERDES6G output buffer

*Table 519 •* **Fields in SERDES6G_OB_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OB_IDLE | 31 | R/W | PCIe support<br>1: idle - force to 0V differential<br>0: Normal mode | 0x0 |
| OB_ENA1V_MODE | 30 | R/W | Output buffer supply voltage<br>1: Set to nominal 1V<br>0: Set to higher voltage | 0x0 |
| OB_POL | 29 | R/W | Polarity of output signal<br>0: Normal<br>1: Inverted | 0x0 |
| OB_POST0 | 28:23 | R/W | Coefficients for 1st Post Cursor (MSB is sign) | 0x00 |
| OB_POST1 | 22:18 | R/W | Coefficients for 2nd Post Cursor (MSB is sign) | 0x00 |
| OB_PREC | 17:13 | R/W | Coefficients for Pre Cursor (MSB is sign) | 0x00 |
| RESERVED | 12:9 | R/W | Must be set to its default. | 0x0 |
| OB_SR_H | 8 | R/W | Half the predriver speed, use for slew rate control<br>0: Disable - slew rate < 60 ps<br>1: Enable - slew rate > 60 ps | 0x0 |
| OB_RESISTOR_CTRL | 7:4 | R/W | Resistor control. Value be taken from RCOMP_STATUS.RCOMP. | 0x0 |
| OB_SR | 3:0 | R/W | Driver speed, fine adjustment of slew rate 30-60ps (if OB_SR_H = 0), 60-140ps (if OB_SR_H = 1) | 0x0 |

### 7.12.8.5 HSIO:SERDES6G_ANA_CFG:SERDES6G_OB_CFG1

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register 1 for SERDES6G output buffer

*Table 520 •* **Fields in SERDES6G_OB_CFG1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OB_ENA_CAS | 8:6 | R/W | Output skew, used for skew adjustment in SGMII mode | 0x0 |
| OB_LEV | 5:0 | R/W | Level of output amplitude<br>0: lowest level<br>63: highest level | 0x00 |

### 7.12.8.6 HSIO:SERDES6G_ANA_CFG:SERDES6G_SER_CFG

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register for SERDES6G serializer

*Table 521* • **Fields in SERDES6G_SER_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 8:4 | R/W | Must be set to its default. | 0x00 |
| SER_ENHYS | 3 | R/W | Enable hysteresis for phase alignment<br>0: Disable hysteresis<br>1: Enable hysteresis | 0x0 |
| RESERVED | 2 | R/W | Must be set to its default. | 0x0 |
| SER_EN_WIN | 1 | R/W | Enable window for phase alignment<br>0: Disable window<br>1: Enable window | 0x0 |
| SER_ENALI | 0 | R/W | Enable phase alignment<br>0: Disable phase alignment<br>1: Enable phase alignment | 0x0 |

### 7.12.8.7 HSIO:SERDES6G_ANA_CFG:SERDES6G_COMMON_CFG

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register for common SERDES6G functions Note: When enabling the facility loop (ena_floop) also the phase alignment in the serializer has to be enabled and configured adequate.

*Table 522* • **Fields in SERDES6G_COMMON_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SYS_RST | 31 | R/W | System reset (low active)<br>0: Apply reset (not self-clearing)<br>1: Reset released | 0x0 |
| ENA_LANE | 18 | R/W | Enable lane<br>0: Disable lane<br>1: Enable line | 0x0 |
| RESERVED | 17:12 | R/W | Must be set to its default. | 0x00 |
| RESERVED | 9:8 | R/W | Must be set to its default. | 0x0 |
| ENA_ELOOP | 11 | R/W | Enable equipment loop<br>0: Disable<br>1: Enable | 0x0 |
| ENA_FLOOP | 10 | R/W | Enable facility loop<br>0: Disable<br>1: Enable | 0x0 |
| HRATE | 7 | R/W | Enable half rate<br>0: Disable<br>1: Enable | 0x1 |
| QRATE | 6 | R/W | Enable quarter rate<br>0: Disable<br>1: Enable | 0x0 |

*Table 522 •* **Fields in SERDES6G_COMMON_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IF_MODE | 5:4 | R/W | Interface mode<br>0: Reserved<br>1: 10-bit mode<br>2: Reserved<br>3: 20-bit mode | 0x1 |

### 7.12.8.8 HSIO:SERDES6G_ANA_CFG:SERDES6G_PLL_CFG

**Parent:** HSIO:SERDES6G_ANA_CFG

**Instances:** 1

Configuration register for SERDES6G RCPLL

*Table 523 •* **Fields in SERDES6G_PLL_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 20 | R/W | Must be set to its default. | 0x0 |
| PLL_ENA_ROT | 18 | R/W | Enable rotation | 0x1 |
| PLL_FSM_CTRL_DATA | 15:8 | R/W | Control data for FSM | 0x00 |
| PLL_FSM_ENA | 7 | R/W | Enable FSM | 0x0 |
| RESERVED | 6:5 | R/W | Must be set to its default. | 0x0 |
| RESERVED | 3 | R/W | Must be set to its default. | 0x0 |
| PLL_ROT_DIR | 2 | R/W | Select rotation direction | 0x0 |
| PLL_ROT_FRQ | 1 | R/W | Select rotation frequency | 0x1 |

## 7.12.9 HSIO:SERDES6G_DIG_CFG

**Parent:** HSIO

**Instances:** 1

Configuration register set for SERDES6G digital BIST and DFT functions.

*Table 524 •* **Registers in SERDES6G_DIG_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SERDES6G_DIG_CFG | 0x00000000 | 1 | SERDES6G Digital Configuration register | Page 409 |
| SERDES6G_MISC_CFG | 0x00000018 | 1 | SERDES6G Misc Configuration | Page 410 |

### 7.12.9.1 HSIO:SERDES6G_DIG_CFG:SERDES6G_DIG_CFG

**Parent:** HSIO:SERDES6G_DIG_CFG

**Instances:** 1

Configuration register for SERDES6G digital functions

*Table 525 •* **Fields in SERDES6G_DIG_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SIGDET_AST | 5:3 | R/W | Signal detect assertion time<br>0: 0 us<br>1: 35 us<br>2: 70 us<br>3: 105 us<br>4: 140 us<br>5..7: reserved | 0x0 |
| SIGDET_DST | 2:0 | R/W | Signal detect de-assertion time<br>0: 0 us<br>1: 250 us<br>2: 350 us<br>3: 450 us<br>4: 550 us<br>5..7: reserved | 0x0 |

### 7.12.9.2 HSIO:SERDES6G_DIG_CFG:SERDES6G_MISC_CFG

**Parent:** HSIO:SERDES6G_DIG_CFG

**Instances:** 1

Configuration register for miscellaneous functions

*Table 526 •* **Fields in SERDES6G_MISC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DES_100FX_CPMD_ENA | 8 | R/W | Enable deserializer cp/md handling for 100fx mode<br>0: Disable<br>1: Enable | 0x0 |
| RX_LPI_MODE_ENA | 5 | R/W | Enable RX-Low-Power feature (Power control by LPI-FSM in connected PCS)<br>0: Disable<br>1: Enable | 0x0 |
| TX_LPI_MODE_ENA | 4 | R/W | Enable TX-Low-Power feature (Power control by LPI-FSM in connected PCS)<br>0: Disable<br>1: Enable | 0x0 |
| RX_DATA_INV_ENA | 3 | R/W | Enable data inversion received from Deserializer<br>0: Disable<br>1: Enable | 0x0 |
| TX_DATA_INV_ENA | 2 | R/W | Enable data inversion sent to Serializer<br>0: Disable<br>1: Enable | 0x0 |

*Table 526 •* **Fields in SERDES6G_MISC_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LANE_RST | 0 | R/W | Lane Reset<br>0: No reset<br>1: Reset (not self-clearing) | 0x0 |

## 7.12.10 HSIO:MCB_SERDES6G_CFG

**Parent:** HSIO

**Instances:** 1

All SERDES6G macros are accessed via the serial Macro Configuration Bus (MCB). Each macro is configured by one MCB Slave. All MCB Slaves are connected in a daisy-chain loop.

*Table 527 •* **Registers in MCB_SERDES6G_CFG**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MCB_SERDES6G_ADDR_CFG | 0x00000000 | 1 | MCB SERDES6G Address Cfg | Page 411 |

### 7.12.10.1 HSIO:MCB_SERDES6G_CFG:MCB_SERDES6G_ADDR_CFG

**Parent:** HSIO:MCB_SERDES6G_CFG

**Instances:** 1

Configuration of SERDES6G MCB Slaves to be accessed

*Table 528 •* **Fields in MCB_SERDES6G_ADDR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SERDES6G_WR_ONE_SHOT | 31 | One-shot | Initiate a write access to marked SERDES6G Slaves<br>0: No write operation pending<br>1: Initiate write to slaves (kept 1 until write operation has finished) | 0x0 |
| SERDES6G_RD_ONE_SHOT | 30 | One-shot | Initiate a read access to marked SERDES6G Slaves<br>0: No read operation pending (read op finished after bit has been set)<br>1: Initiate a read access (kept 1 until read operation has finished) | 0x0 |
| SERDES6G_ADDR | 15:0 | R/W | Activation vector for SERDES6G-Slaves, one-hot coded, each bit is related to one macro, e.g. bit 0 enables/disables access to macro No. 0<br>0: Disable macro access via MCB<br>1: Enable macro access via MCB | 0xFFFF |

# 7.13 DEV_GMII

*Table 529* • **Register Groups in DEV_GMII**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PORT_MODE | 0x00000000 | 1 | | Page 412 |
| MAC_CFG_STATUS | 0x0000000C | 1 | | Page 413 |

## 7.13.1 DEV_GMII:PORT_MODE

**Parent:** DEV_GMII

**Instances:** 1

*Table 530* • **Registers in PORT_MODE**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CLOCK_CFG | 0x00000000 | 1 | | Page 412 |
| PORT_MISC | 0x00000004 | 1 | | Page 412 |

### 7.13.1.1 DEV_GMII:PORT_MODE:CLOCK_CFG

**Parent:** DEV_GMII:PORT_MODE

**Instances:** 1

*Table 531* • **Fields in CLOCK_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAC_TX_RST | 3 | R/W | | 0x1 |
| MAC_RX_RST | 2 | R/W | | 0x1 |
| PORT_RST | 1 | R/W | | 0x1 |
| PHY_RST | 0 | R/W | | 0x1 |

### 7.13.1.2 DEV_GMII:PORT_MODE:PORT_MISC

**Parent:** DEV_GMII:PORT_MODE

**Instances:** 1

*Table 532 •* **Fields in PORT_MISC**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FWD_PAUSE_ENA | 3 | R/W | Forward pause frames (EtherType = 0x8808, opcode = 0x0001). The reaction to incoming pause frames is controlled independently of FWD_PAUSE_ENA. | 0x0 |
| FWD_CTRL_ENA | 2 | R/W | Forward MAC control frames excluding pause frames (EtherType = 0x8808, opcode different from 0x0001). | 0x0 |
| GMII_LOOP_ENA | 1 | R/W | Loop GMII transmit data directly into receive path. | 0x0 |
| DEV_LOOP_ENA | 0 | R/W | Loop the device bus through this port. The MAC is potentially bypassed. | 0x0 |

## 7.13.2 DEV_GMII:MAC_CFG_STATUS

**Parent:** DEV_GMII

**Instances:** 1

The 1G MAC module contains configuration and status registers related to the MAC module of the 1G Device.

*Table 533 •* **Registers in MAC_CFG_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MAC_ENA_CFG | 0x00000000 | 1 | Mode Configuration Register | Page 414 |
| MAC_MODE_CFG | 0x00000004 | 1 | Mode Configuration Register | Page 414 |
| MAC_MAXLEN_CFG | 0x00000008 | 1 | Max Length Configuration Register | Page 414 |
| MAC_TAGS_CFG | 0x0000000C | 1 | VLAN/Service tag configuration register | Page 415 |
| MAC_ADV_CHK_CFG | 0x00000010 | 1 | Advanced Check Feature Configuration Register | Page 416 |
| MAC_IFG_CFG | 0x00000014 | 1 | Inter Frame Gap Configuration Register | Page 416 |
| MAC_HDX_CFG | 0x00000018 | 1 | Half Duplex Configuration Register | Page 417 |
| MAC_FC_CFG | 0x00000020 | 1 | MAC Flow Control Configuration Register | Page 418 |
| MAC_FC_MAC_LOW_CFG | 0x00000024 | 1 | MAC Flow Control Configuration Register | Page 419 |
| MAC_FC_MAC_HIGH_CFG | 0x00000028 | 1 | MAC Flow Control Configuration Register | Page 419 |

*Table 533 •* **Registers in MAC_CFG_STATUS *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MAC_STICKY | 0x0000002C | 1 | Sticky Bit Register | Page 419 |

### 7.13.2.1 DEV_GMII:MAC_CFG_STATUS:MAC_ENA_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 534 •* **Fields in MAC_ENA_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_ENA | 4 | R/W | Receiver Module Enable.<br>'0': Receiver Module Disabled<br>'1': Receiver Module Enabled | 0x0 |
| TX_ENA | 0 | R/W | Transmitter Module Enable.<br>'0': Transmitter Module Disabled<br>'1': Transmitter Module Enabled | 0x0 |

### 7.13.2.2 DEV_GMII:MAC_CFG_STATUS:MAC_MODE_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 535 •* **Fields in MAC_MODE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GIGA_MODE_ENA | 4 | R/W | Enables 1 Gbps mode.<br>'0': 10/100 Mbps mode<br>'1': 1 Gbps mode. Note: FDX_ENA must also be set. | 0x1 |
| FDX_ENA | 0 | R/W | Enables Full Duplex:<br>'0': Half Duplex<br>'1': Full duplex.<br><br>Note: Full duplex MUST be selected if GIGA_MODE is enabled. | 0x1 |

### 7.13.2.3 DEV_GMII:MAC_CFG_STATUS:MAC_MAXLEN_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 536 •* **Fields in MAC_MAXLEN_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAX_LEN | 15:0 | R/W | When set, single tagged frames are allowed to be 4 bytes longer than the MAC_MAXLEN_CFG configuration and double tagged frames are allowed to be 8 bytes longer. Single tagged frames are adjusted if VLAN_AWR_ENA is also set. Double tagged frames are adjusted if both VLAN_AWR_ENA and VLAN_DBL_AWR_ENA are set. | 0x05EE |

### 7.13.2.4 DEV_GMII:MAC_CFG_STATUS:MAC_TAGS_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

The MAC can be configured to accept 0, 1 and 2 tags and the TAG value can be user-defined.

*Table 537 •* **Fields in MAC_TAGS_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TAG_ID | 31:16 | R/W | This field defines which EtherTypes are recognized as a VLAN TPID - besides 0x8100. The value is used for all tag positions. I.e. a double tagged frame can have the following tag values: (TAG1,TAG2):<br>( 0x8100, 0x8100 )<br>( 0x8100, TAG_ID )<br>( TAG_ID, 0x8100 ) or<br>( TAG_ID, TAG_ID )<br><br>Single tagged frame can have the following TPID values: 0x8100 or TAG_ID. | 0x8100 |
| VLAN_DBL_AWR_ENA | 1 | R/W | If set, double tagged frames are subject to length adjustments (VLAN_LEN_AWR_ENA). VLAN_AWR_ENA must be set when VLAN_DBL_AWR_ENA is set.<br>'0': The MAC does not look for inner tags.<br>'1': The MAC accepts inner tags with TPID=0x8100 or TAG_ID. | 0x0 |

*Table 537 •* **Fields in MAC_TAGS_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| VLAN_AWR_ENA | 0 | R/W | If set, single tagged frames are subject to length adjustments (VLAN_LEN_AWR_ENA).<br>'0': The MAC does not look for any tags.<br>'1': The MAC accepts outer tags with TPID=0x8100 or TAG_ID. | 0x0 |
| VLAN_LEN_AWR_ENA | 2 | R/W | When set, single tagged frames are allowed to be 4 bytes longer than the MAC_MAXLEN_CFG configuration and double tagged frames are allowed to be 8 bytes longer. Single tagged frames are adjusted if VLAN_AWR_ENA is also set. Double tagged frames are adjusted if both VLAN_AWR_ENA and VLAN_DBL_AWR_ENA are set. | 0x1 |

### 7.13.2.5 DEV_GMII:MAC_CFG_STATUS:MAC_ADV_CHK_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 538 •* **Fields in MAC_ADV_CHK_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LEN_DROP_ENA | 0 | R/W | Length Drop Enable:<br>Configures the Receive Module to drop frames in reference to in-range and out-of-range errors:<br>'0': Length Drop Disabled<br>'1': Length Drop Enabled. | 0x0 |

### 7.13.2.6 DEV_GMII:MAC_CFG_STATUS:MAC_IFG_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 539 •* **Fields in MAC_IFG_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_IFG | 12:8 | R/W | Used to adjust the duration of the inter-frame gap in the Tx direction and must be set according to the speed and duplex settings.<br>10/100 Mbps, HDX, FDX 0x19, 0x13<br>1000 Mbps: 0x07. | 0x07 |

*Table 539 •* **Fields in MAC_IFG_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_IFG2 | 7:4 | R/W | Used to adjust the duration of the second part of the inter-frame gap in the Rx direction and must be set according to the speed and duplex settings.<br>10/100 Mbps, HDX, FDX: 0x8, 0xB<br>1000 Mbps: 0x1. | 0x1 |
| RX_IFG1 | 3:0 | R/W | Used to adjust the duration of the first part of the inter-frame gap in the Rx direction and must be set according to the speed settings.<br>10/100 Mbps: 0x7<br>1000 Mbps: 0x5. | 0x5 |

### 7.13.2.7 DEV_GMII:MAC_CFG_STATUS:MAC_HDX_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 540 •* **Fields in MAC_HDX_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WEXC_DIS | 24 | R/W | Determines whether the MAC backs off after an excessive collision has occurred. If set, back off is disabled after excessive collisions.<br>'0': Back off after excessive collisions<br>'1': Don't back off after excessive collisions | 0x0 |
| SEED | 23:16 | R/W | Seed value loaded into the PRBS of the MAC.<br>Used to prevent excessive collision events. | 0x00 |
| SEED_LOAD | 12 | R/W | Load SEED value into PRNG register. A SEED value is loaded into the PRNG register of the MAC, when SEED_LOAD is asserted. After a load, the SEED_LOAD must be deasserted.<br>'0': Do not load SEED value<br>'1': Load SEED value. | 0x0 |

*Table 540 •* **Fields in MAC_HDX_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RETRY_AFTER_EXC_COL_ENA | 8 | R/W | This bit is used to setup the MAC to retransmit a frame after an early collision even though 16 (or more) early collisions have occurred.<br>'0': A frame is discarded and counted as an excessive collision if 16 collisions occur for this frame.<br>'1': The MAC retransmits a frame after an early collision, regardless of the number of previous early collisions. The backoff sequence is reset after every 16 collisions. | 0x0 |
| LATE_COL_POS | 6:0 | R/W | Adjustment of early/late collision boundary:<br>This bitgroup is used to adjust the MAC so that a collision on a shared transmission medium before bit 512 is handled as an early collision, whereas a collision after bit 512 is handled as a late collision, i.e. no retransmission is performed. | 0x43 |

### 7.13.2.8 DEV_GMII:MAC_CFG_STATUS:MAC_FC_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 541 •* **Fields in MAC_FC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ZERO_PAUSE_ENA | 18 | R/W | If set, a zero-delay pause frame is transmitted when flow control is deasserted.<br>'0': Don't send zero pause frame.<br>'1': Send zero pause frame. | 0x0 |
| TX_FC_ENA | 17 | R/W | When set the MAC will send pause control frames in the Tx direction.<br>'0': Don't send pause control frames<br>'1': Send pause control frames | 0x0 |
| RX_FC_ENA | 16 | R/W | When set the MAC obeys received pause control frames<br>'0': Don't obey received pause control frames<br>'1': Obey received pause control frames. | 0x0 |

*Table 541 •* **Fields in MAC_FC_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAUSE_VAL_CFG | 15:0 | R/W | Pause timer value inserted in generated pause frames.<br>0: Insert timer value 0 in TX pause frame.<br>...<br>N: Insert timer value N in TX pause frame. | 0x0000 |
| FC_LATENCY_CFG | 24:19 | R/W | Accepted reaction time for link partner after the port has transmitted a pause frame. Frames starting after this latency are aborted. Unit is 64 byte times. A value of 63 disables the feature. For proper flow control operation, use FC_LATENCY_CFG = 7. | 0x03 |

### 7.13.2.9 DEV_GMII:MAC_CFG_STATUS:MAC_FC_MAC_LOW_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 542 •* **Fields in MAC_FC_MAC_LOW_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAC_LOW | 23:0 | R/W | Lower three bytes in the SMAC in generated flow control frames.<br><br>0xNNN: Lower three DMAC bytes | 0x000000 |

### 7.13.2.10 DEV_GMII:MAC_CFG_STATUS:MAC_FC_MAC_HIGH_CFG

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

*Table 543 •* **Fields in MAC_FC_MAC_HIGH_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAC_HIGH | 23:0 | R/W | Higher three bytes in the SMAC in generated flow control frames.<br>0xNNN: Higher three DMAC bytes | 0x000000 |

### 7.13.2.11 DEV_GMII:MAC_CFG_STATUS:MAC_STICKY

**Parent:** DEV_GMII:MAC_CFG_STATUS

**Instances:** 1

Clear the sticky bits by writing a '0' in the relevant bitgroups (writing a '1' sets the bit)!.

*Table 544 •* **Fields in MAC_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_IPG_SHRINK_STICKY | 9 | Sticky | Sticky bit indicating that an inter packet gap shrink was detected (IPG < 12 bytes). | 0x0 |
| RX_PREAM_SHRINK_STICKY | 8 | Sticky | Sticky bit indicating that a preamble shrink was detected (preamble < 8 bytes).<br>'0': no preamble shrink was detected<br>'1': a preamble shrink was detected one or more times<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| RX_CARRIER_EXT_STICKY | 7 | Sticky | Sticky bit indicating that a carrier extend was detected.<br>'0': no carrier extend was detected<br>'1': one or more carrier extends were detected<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| RX_CARRIER_EXT_ERR_STICKY | 6 | Sticky | Sticky bit indicating that a carrier extend error was detected.<br>'0': no carrier extend error was detected<br>'1': one or more carrier extend errors were detected<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| RX_JUNK_STICKY | 5 | Sticky | Sticky bit indicating that junk was received (bytes not recognized as a frame).<br>'0': no junk was received<br>'1': junk was received one or more times<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_RETRANSMIT_STICKY | 4 | Sticky | Sticky bit indicating that the transmit MAC asked the host for a frame retransmission.<br>'0': no tx retransmission was initiated<br>'1': one or more tx retransmissions were initiated<br>Bit is cleared by writing a '1' to this position. | 0x0 |

*Table 544 •* **Fields in MAC_STICKY *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_JAM_STICKY | 3 | Sticky | Sticky bit indicating that the transmit host issued a jamming signal.<br>'0': the transmit host issued no jamming signal<br>'1': the transmit host issued one or more jamming signals<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_FIFO_OFLW_STICKY | 2 | Sticky | Sticky bit indicating that the MAC transmit FIFO has overrun. | 0x0 |
| TX_FRM_LEN_OVR_STICKY | 1 | Sticky | Sticky bit indicating that the transmit frame length has overrun. I.e. a frame longer than 64K occurred.<br>'0': no tx frame length error occurred<br>'1': one or more tx frames length errors occurred<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_ABORT_STICKY | 0 | Sticky | Sticky bit indicating that the transmit host initiated abort was executed. | 0x0 |

# 7.14 DEV

*Table 545 •* **Register Groups in DEV**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PORT_MODE | 0x00000004 | 1 | | Page 421 |
| MAC_CFG_STATUS | 0x00000010 | 1 | | Page 423 |
| PCS1G_CFG_STATUS | 0x00000040 | 1 | PCS 1G Configuration Status Registers | Page 430 |
| PCS1G_TSTPAT_CFG_STATUS | 0x00000084 | 1 | PCS1G Testpattern Configuration and Status Registers | Page 438 |
| PCS_FX100_CONFIGURATION | 0x0000008C | 1 | PCS FX100 Configuration Registers | Page 440 |
| PCS_FX100_STATUS | 0x00000090 | 1 | PCS FX100 Status Registers | Page 441 |

## 7.14.1 DEV:PORT_MODE

**Parent:** DEV

**Instances:** 1

*Table 546* • **Registers in PORT_MODE**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CLOCK_CFG | 0x00000000 | 1 | | Page 422 |
| PORT_MISC | 0x00000004 | 1 | | Page 422 |

### 7.14.1.1 DEV:PORT_MODE:CLOCK_CFG

**Parent:** DEV:PORT_MODE

**Instances:** 1

*Table 547* • **Fields in CLOCK_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAC_TX_RST | 7 | R/W | | 0x1 |
| MAC_RX_RST | 6 | R/W | | 0x1 |
| PCS_TX_RST | 5 | R/W | | 0x1 |
| PCS_RX_RST | 4 | R/W | | 0x1 |
| PORT_RST | 3 | R/W | | 0x1 |
| PHY_RST | 2 | R/W | Only applicable to ports 10 and 11. | 0x1 |
| LINK_SPEED | 1:0 | R/W | Selects the link speed. For ports 10 and 11, LINK_SPEED is ignored when DEV_IF_CFG.GMII_DIS is cleared.<br>0: No link<br>1: 1000 Mbps<br>2: 100 Mbps<br>3: 10 Mbps | 0x0 |

### 7.14.1.2 DEV:PORT_MODE:PORT_MISC

**Parent:** DEV:PORT_MODE

**Instances:** 1

*Table 548* • **Fields in PORT_MISC**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FWD_PAUSE_ENA | 2 | R/W | Forward pause frames (EtherType = 0x8808, opcode = 0x0001). The reaction to incoming pause frames is controlled independently of FWD_PAUSE_ENA. | 0x0 |

*Table 548 •* **Fields in PORT_MISC *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FWD_CTRL_ENA | 1 | R/W | Forward MAC control frames excluding pause frames (EtherType = 0x8808, opcode different from 0x0001). | 0x0 |
| DEV_LOOP_ENA | 0 | R/W | Loop the device bus through this port. The MAC is potentially bypassed. | 0x0 |

## 7.14.2 DEV:MAC_CFG_STATUS

**Parent:** DEV

**Instances:** 1

The 1G MAC module contains configuration and status registers related to the MAC module of the 1G Device.

*Table 549 •* **Registers in MAC_CFG_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MAC_ENA_CFG | 0x00000000 | 1 | Mode Configuration Register | Page 423 |
| MAC_MODE_CFG | 0x00000004 | 1 | Mode Configuration Register | Page 424 |
| MAC_MAXLEN_CFG | 0x00000008 | 1 | Max Length Configuration Register | Page 424 |
| MAC_TAGS_CFG | 0x0000000C | 1 | VLAN/Service tag configuration register | Page 425 |
| MAC_ADV_CHK_CFG | 0x00000010 | 1 | Advanced Check Feature Configuration Register | Page 426 |
| MAC_IFG_CFG | 0x00000014 | 1 | Inter Frame Gap Configuration Register | Page 426 |
| MAC_HDX_CFG | 0x00000018 | 1 | Half Duplex Configuration Register | Page 426 |
| MAC_FC_CFG | 0x00000020 | 1 | MAC Flow Control Configuration Register | Page 427 |
| MAC_FC_MAC_LOW_CFG | 0x00000024 | 1 | MAC Flow Control Configuration Register | Page 428 |
| MAC_FC_MAC_HIGH_CFG | 0x00000028 | 1 | MAC Flow Control Configuration Register | Page 428 |
| MAC_STICKY | 0x0000002C | 1 | Sticky Bit Register | Page 429 |

### 7.14.2.1 DEV:MAC_CFG_STATUS:MAC_ENA_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

*Table 550 •* **Fields in MAC_ENA_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_ENA | 4 | R/W | Receiver Module Enable.<br>'0': Receiver Module Disabled<br>'1': Receiver Module Enabled | 0x0 |
| TX_ENA | 0 | R/W | Transmitter Module Enable.<br>'0': Transmitter Module Disabled<br>'1': Transmitter Module Enabled | 0x0 |

### 7.14.2.2 DEV:MAC_CFG_STATUS:MAC_MODE_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

*Table 551 •* **Fields in MAC_MODE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GIGA_MODE_ENA | 4 | R/W | Enables 1 Gbps mode.<br>'0': 10/100 Mbps mode<br>'1': 1 Gbps mode. Note: FDX_ENA must also be set. | 0x1 |
| FDX_ENA | 0 | R/W | Enables Full Duplex:<br>'0': Half Duplex<br>'1': Full duplex.<br><br>Note: Full duplex MUST be selected if GIGA_MODE is enabled. | 0x1 |

### 7.14.2.3 DEV:MAC_CFG_STATUS:MAC_MAXLEN_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

*Table 552 •* **Fields in MAC_MAXLEN_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAX_LEN | 15:0 | R/W | When set, single tagged frames are allowed to be 4 bytes longer than the MAC_MAXLEN_CFG configuration and double tagged frames are allowed to be 8 bytes longer. Single tagged frames are adjusted if VLAN_AWR_ENA is also set. Double tagged frames are adjusted if both VLAN_AWR_ENA and VLAN_DBL_AWR_ENA are set. | 0x05EE |

### 7.14.2.4 DEV:MAC_CFG_STATUS:MAC_TAGS_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

The MAC can be configured to accept 0, 1 and 2 tags and the TAG value can be user-defined.

***Table 553 •*** **Fields in MAC_TAGS_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TAG_ID | 31:16 | R/W | This field defines which EtherTypes are recognized as a VLAN TPID - besides 0x8100. The value is used for all tag positions. I.e. a double tagged frame can have the following tag values: (TAG1,TAG2):<br>( 0x8100, 0x8100 )<br>( 0x8100, TAG_ID )<br>( TAG_ID, 0x8100 ) or<br>( TAG_ID, TAG_ID )<br><br>Single tagged frame can have the following TPID values: 0x8100 or TAG_ID. | 0x8100 |
| VLAN_DBL_AWR_ENA | 1 | R/W | If set, double tagged frames are subject to length adjustments (VLAN_LEN_AWR_ENA). VLAN_AWR_ENA must be set when VLAN_DBL_AWR_ENA is set.<br>'0': The MAC does not look for inner tags.<br>'1': The MAC accepts inner tags with TPID=0x8100 or TAG_ID. | 0x0 |
| VLAN_AWR_ENA | 0 | R/W | If set, single tagged frames are subject to length adjustments (VLAN_LEN_AWR_ENA).<br>'0': The MAC does not look for any tags.<br>'1': The MAC accepts outer tags with TPID=0x8100 or TAG_ID. | 0x0 |
| VLAN_LEN_AWR_ENA | 2 | R/W | When set, single tagged frames are allowed to be 4 bytes longer than the MAC_MAXLEN_CFG configuration and double tagged frames are allowed to be 8 bytes longer. Single tagged frames are adjusted if VLAN_AWR_ENA is also set. Double tagged frames are adjusted if both VLAN_AWR_ENA and VLAN_DBL_AWR_ENA are set. | 0x1 |

### 7.14.2.5 DEV:MAC_CFG_STATUS:MAC_ADV_CHK_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

***Table 554 •*** **Fields in MAC_ADV_CHK_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LEN_DROP_ENA | 0 | R/W | Length Drop Enable:<br>Configures the Receive Module to drop frames in reference to in-range and out-of-range errors:<br>'0': Length Drop Disabled<br>'1': Length Drop Enabled. | 0x0 |

### 7.14.2.6 DEV:MAC_CFG_STATUS:MAC_IFG_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

***Table 555 •*** **Fields in MAC_IFG_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_IFG | 12:8 | R/W | Used to adjust the duration of the inter-frame gap in the Tx direction and must be set according to the speed and duplex settings.<br>10/100 Mbps, HDX, FDX 0x19, 0x13<br>1000 Mbps: 0x07. | 0x07 |
| RX_IFG2 | 7:4 | R/W | Used to adjust the duration of the second part of the inter-frame gap in the Rx direction and must be set according to the speed and duplex settings.<br>10/100 Mbps, HDX, FDX: 0x8, 0xB<br>1000 Mbps: 0x1. | 0x1 |
| RX_IFG1 | 3:0 | R/W | Used to adjust the duration of the first part of the inter-frame gap in the Rx direction and must be set according to the speed settings.<br>10/100 Mbps: 0x7<br>1000 Mbps: 0x5. | 0x5 |

### 7.14.2.7 DEV:MAC_CFG_STATUS:MAC_HDX_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

*Table 556 •* **Fields in MAC_HDX_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WEXC_DIS | 24 | R/W | Determines whether the MAC backs off after an excessive collision has occurred. If set, back off is disabled after excessive collisions.<br>'0': Back off after excessive collisions<br>'1': Don't back off after excessive collisions | 0x0 |
| SEED | 23:16 | R/W | Seed value loaded into the PRBS of the MAC.<br>Used to prevent excessive collision events. | 0x00 |
| SEED_LOAD | 12 | R/W | Load SEED value into PRNG register. A SEED value is loaded into the PRNG register of the MAC, when SEED_LOAD is asserted. After a load, the SEED_LOAD must be deasserted.<br>'0': Do not load SEED value<br>'1': Load SEED value. | 0x0 |
| RETRY_AFTER_EXC_COL_ENA | 8 | R/W | This bit is used to setup the MAC to retransmit a frame after an early collision even though 16 (or more) early collisions have occurred.<br>'0': A frame is discarded and counted as an excessive collision if 16 collisions occur for this frame.<br>'1': The MAC retransmits a frame after an early collision, regardless of the number of previous early collisions. The backoff sequence is reset after every 16 collisions. | 0x0 |
| LATE_COL_POS | 6:0 | R/W | Adjustment of early/late collision boundary:<br>This bitgroup is used to adjust the MAC so that a collision on a shared transmission medium before bit 512 is handled as an early collision, whereas a collision after bit 512 is handled as a late collision, i.e. no retransmission is performed. | 0x43 |

### 7.14.2.8 DEV:MAC_CFG_STATUS:MAC_FC_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

*Table 557 •* **Fields in MAC_FC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ZERO_PAUSE_ENA | 18 | R/W | If set, a zero-delay pause frame is transmitted when flow control is deasserted.<br>'0': Don't send zero pause frame.<br>'1': Send zero pause frame. | 0x0 |
| TX_FC_ENA | 17 | R/W | When set the MAC will send pause control frames in the Tx direction.<br>'0': Don't send pause control frames<br>'1': Send pause control frames | 0x0 |
| RX_FC_ENA | 16 | R/W | When set the MAC obeys received pause control frames<br>'0': Don't obey received pause control frames<br>'1': Obey received pause control frames. | 0x0 |
| PAUSE_VAL_CFG | 15:0 | R/W | Pause timer value inserted in generated pause frames.<br>0: Insert timer value 0 in TX pause frame.<br>...<br>N: Insert timer value N in TX pause frame. | 0x0000 |
| FC_LATENCY_CFG | 24:19 | R/W | Accepted reaction time for link partner after the port has transmitted a pause frame. Frames starting after this latency are aborted. Unit is 64 byte times. A value of 63 disables the feature. For proper flow control operation, use FC_LATENCY_CFG = 7. | 0x03 |

### 7.14.2.9 DEV:MAC_CFG_STATUS:MAC_FC_MAC_LOW_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

*Table 558 •* **Fields in MAC_FC_MAC_LOW_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAC_LOW | 23:0 | R/W | Lower three bytes in the SMAC in generated flow control frames.<br><br>0xNNN: Lower three DMAC bytes | 0x000000 |

### 7.14.2.10 DEV:MAC_CFG_STATUS:MAC_FC_MAC_HIGH_CFG

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

***Table 559 •*** **Fields in MAC_FC_MAC_HIGH_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAC_HIGH | 23:0 | R/W | Higher three bytes in the SMAC in generated flow control frames.<br>0xNNN: Higher three DMAC bytes | 0x000000 |

### 7.14.2.11 DEV:MAC_CFG_STATUS:MAC_STICKY

**Parent:** DEV:MAC_CFG_STATUS

**Instances:** 1

Clear the sticky bits by writing a '0' in the relevant bitgroups (writing a '1' sets the bit)!.

***Table 560 •*** **Fields in MAC_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_IPG_SHRINK_STICKY | 9 | Sticky | Sticky bit indicating that an inter packet gap shrink was detected (IPG < 12 bytes). | 0x0 |
| RX_PREAM_SHRINK_STICKY | 8 | Sticky | Sticky bit indicating that a preamble shrink was detected (preamble < 8 bytes).<br>'0': no preamble shrink was detected<br>'1': a preamble shrink was detected one or more times<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| RX_CARRIER_EXT_STICKY | 7 | Sticky | Sticky bit indicating that a carrier extend was detected.<br>'0': no carrier extend was detected<br>'1': one or more carrier extends were detected<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| RX_CARRIER_EXT_ERR_STICKY | 6 | Sticky | Sticky bit indicating that a carrier extend error was detected.<br>'0': no carrier extend error was detected<br>'1': one or more carrier extend errors were detected<br>Bit is cleared by writing a '1' to this position. | 0x0 |

*Table 560 •* **Fields in MAC_STICKY** *(continued)*

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_JUNK_STICKY | 5 | Sticky | Sticky bit indicating that junk was received (bytes not recognized as a frame).<br>'0': no junk was received<br>'1': junk was received one or more times<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_RETRANSMIT_STICKY | 4 | Sticky | Sticky bit indicating that the transmit MAC asked the host for a frame retransmission.<br>'0': no tx retransmission was initiated<br>'1': one or more tx retransmissions were initiated<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_JAM_STICKY | 3 | Sticky | Sticky bit indicating that the transmit host issued a jamming signal.<br>'0': the transmit host issued no jamming signal<br>'1': the transmit host issued one or more jamming signals<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_FIFO_OFLW_STICKY | 2 | Sticky | Sticky bit indicating that the MAC transmit FIFO has overrun. | 0x0 |
| TX_FRM_LEN_OVR_STICKY | 1 | Sticky | Sticky bit indicating that the transmit frame length has overrun. I.e. a frame longer than 64K occurred.<br>'0': no tx frame length error occurred<br>'1': one or more tx frames length errors occurred<br>Bit is cleared by writing a '1' to this position. | 0x0 |
| TX_ABORT_STICKY | 0 | Sticky | Sticky bit indicating that the transmit host initiated abort was executed. | 0x0 |

## 7.14.3 DEV:PCS1G_CFG_STATUS

**Parent:** DEV

**Instances:** 1

Configuration and status register set for PCS1G

*Table 561 •* **Registers in PCS1G_CFG_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PCS1G_CFG | 0x00000000 | 1 | PCS1G Configuration | Page 431 |
| PCS1G_MODE_CFG | 0x00000004 | 1 | PCS1G Mode Configuration | Page 432 |
| PCS1G_SD_CFG | 0x00000008 | 1 | PCS1G Signal Detect Configuration | Page 432 |
| PCS1G_ANEG_CFG | 0x0000000C | 1 | PCS1G Aneg Configuration | Page 433 |
| PCS1G_ANEG_NP_CFG | 0x00000010 | 1 | PCS1G Aneg Next Page Configuration | Page 433 |
| PCS1G_LB_CFG | 0x00000014 | 1 | PCS1G Loopback Configuration | Page 434 |
| PCS1G_ANEG_STATUS | 0x00000020 | 1 | PCS1G ANEG Status Register | Page 434 |
| PCS1G_ANEG_NP_STATUS | 0x00000024 | 1 | PCS1G Aneg Next Page Status Register | Page 435 |
| PCS1G_LINK_STATUS | 0x00000028 | 1 | PCS1G link status | Page 435 |
| PCS1G_LINK_DOWN_CNT | 0x0000002C | 1 | PCS1G link down counter | Page 436 |
| PCS1G_STICKY | 0x00000030 | 1 | PCS1G sticky register | Page 436 |
| PCS1G_LPI_CFG | 0x00000038 | 1 | PCS1G Low Power Idle Configuration | Page 437 |
| PCS1G_LPI_WAKE_ERROR_CNT | 0x0000003C | 1 | PCS1G wake error counter | Page 437 |
| PCS1G_LPI_STATUS | 0x00000040 | 1 | PCS1G Low Power Idle Status | Page 437 |

### 7.14.3.1 DEV:PCS1G_CFG_STATUS:PCS1G_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G main configuration register

*Table 562 •* **Fields in PCS1G_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_STATUS_TYPE | 4 | R/W | Set type of link_status indication at CPU-System<br>0: Sync_status (from PCS synchronization state machine)<br>1: Bit 15 of PCS1G_ANEG_STATUS.lp_adv_ability (Link up/down) | 0x0 |

*Table 562 •* **Fields in PCS1G_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PCS_ENA | 0 | R/W | PCS enable<br>0: Disable PCS<br>1: Enable PCS | 0x0 |

### 7.14.3.2 DEV:PCS1G_CFG_STATUS:PCS1G_MODE_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G mode configuration

*Table 563 •* **Fields in PCS1G_MODE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UNIDIR_MODE_ENA | 4 | R/W | Unidirectional mode enable. Implementation of 802.3, Clause 66. When asserted, this enables MAC to transmit data independent of the state of the receive link.<br>0: Unidirectional mode disabled<br>1: Unidirectional mode enabled | 0x0 |
| SGMII_MODE_ENA | 0 | R/W | Selection of PCS operation<br>0: PCS is used in SERDES mode<br>1: PCS is used in SGMII mode. Configuration bit PCS1G_ANEG_CFG.SW_RESOLVE_ENA must be set additionally | 0x1 |

### 7.14.3.3 DEV:PCS1G_CFG_STATUS:PCS1G_SD_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G signal_detect configuration

*Table 564 •* **Fields in PCS1G_SD_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SD_SEL | 8 | R/W | Signal detect selection (select input for internal signal_detect line)<br>0: Select signal_detect line from hardmacro<br>1: Select external signal_detect line | 0x0 |

*Table 564 •* **Fields in PCS1G_SD_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SD_POL | 4 | R/W | Signal detect polarity: The signal level on signal_detect input pin must be equal to SD_POL to indicate signal detection (SD_ENA must be set)<br>0: Signal Detect input pin must be '0' to indicate a signal detection<br>1: Signal Detect input pin must be '1' to indicate a signal detection | 0x1 |
| SD_ENA | 0 | R/W | Signal Detect Enable<br>0: The Signal Detect input pin is ignored. The PCS assumes an active Signal Detect at all times<br>1: The Signal Detect input pin is used to determine if a signal is detected | 0x1 |

### 7.14.3.4 DEV:PCS1G_CFG_STATUS:PCS1G_ANEG_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G Auto-negotiation configuration register

*Table 565 •* **Fields in PCS1G_ANEG_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ADV_ABILITY | 31:16 | R/W | Advertised Ability Register: Holds the capabilities of the device as described IEEE 802.3, Clause 37. If SGMII mode is selected (PCS1G_MODE_CFG.SGMII_MODE_ENA = 1), SW_RESOLVE_ENA must be set. | 0x0000 |
| SW_RESOLVE_ENA | 8 | R/W | Software Resolve Abilities<br>0: If Auto Negotiation fails (no matching HD or FD capabilities) the link is disabled<br>1: The result of an Auto Negotiation is ignored - the link can be setup via software. This bit must be set in SGMII mode. | 0x0 |
| ANEG_RESTART_ONE_SHOT | 1 | One-shot | Auto Negotiation Restart<br>0: No action<br>1: Restart Auto Negotiation | 0x0 |
| ANEG_ENA | 0 | R/W | Auto Negotiation Enable<br>0: Auto Negotiation Disabled<br>1: Auto Negotiation Enabled | 0x0 |

### 7.14.3.5 DEV:PCS1G_CFG_STATUS:PCS1G_ANEG_NP_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G Auto-negotiation configuration register for next-page function

***Table 566 •*** **Fields in PCS1G_ANEG_NP_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NP_TX | 31:16 | R/W | Next page register: Holds the next-page information as described in IEEE 802.3, Clause 37 | 0x0000 |
| NP_LOADED_ONE_SHOT | 0 | One-shot | Next page loaded<br>0: next page is free and can be loaded<br>1: next page register has been filled (to be set after np_tx has been filled) | 0x0 |

### 7.14.3.6 DEV:PCS1G_CFG_STATUS:PCS1G_LB_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G Loop-Back configuration register

***Table 567 •*** **Fields in PCS1G_LB_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TBI_HOST_LB_ENA | 0 | R/W | Loops data in PCS (TBI side) from egress direction to ingress direction. The Rx clock is automatically set equal to the Tx clock<br>0: TBI Loopback Disabled<br>1:TBI Loopback Enabled | 0x0 |

### 7.14.3.7 DEV:PCS1G_CFG_STATUS:PCS1G_ANEG_STATUS

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G Auto-negotiation status register

***Table 568 •*** **Fields in PCS1G_ANEG_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LP_ADV_ABILITY | 31:16 | R/O | Advertised abilities from link partner as described in IEEE 802.3, Clause 37 | 0x0000 |
| PR | 4 | R/O | Resolve priority<br>0: ANEG is in progress<br>1: ANEG nearly finished - priority can be resolved (via software) | 0x0 |

*Table 568 •* **Fields in PCS1G_ANEG_STATUS *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAGE_RX_STICKY | 3 | Sticky | Status indicating whether a new page has been received.<br>0: No new page received<br>1: New page received<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| ANEG_COMPLETE | 0 | R/O | Auto Negotiation Complete<br>0: No Auto Negotiation has been completed<br>1: Indicates that an Auto Negotiation has completed successfully | 0x0 |

### 7.14.3.8 DEV:PCS1G_CFG_STATUS:PCS1G_ANEG_NP_STATUS

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G Auto-negotiation next page status register

*Table 569 •* **Fields in PCS1G_ANEG_NP_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LP_NP_RX | 31:16 | R/O | Next page ability register from link partner as described in IEEE 802.3, Clause 37 | 0x0000 |

### 7.14.3.9 DEV:PCS1G_CFG_STATUS:PCS1G_LINK_STATUS

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G link status register

*Table 570 •* **Fields in PCS1G_LINK_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SIGNAL_DETECT | 8 | R/O | Indicates whether or not the selected Signal Detect input line is asserted<br>0: No signal detected<br>1: Signal detected | 0x0 |
| LINK_STATUS | 4 | R/O | Indicates whether the link is up or down. A link is up when ANEG state machine is in state LINK_OK or AN_DISABLE_LINK_OK<br>0: Link down<br>1: Link up | 0x0 |

*Table 570 •* **Fields in PCS1G_LINK_STATUS *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SYNC_STATUS | 0 | R/O | Indicates if PCS has successfully synchronized<br>0: PCS is out of sync<br>1: PCS has synchronized | 0x0 |

### 7.14.3.10 DEV:PCS1G_CFG_STATUS:PCS1G_LINK_DOWN_CNT

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G link down counter register

*Table 571 •* **Fields in PCS1G_LINK_DOWN_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_DOWN_CNT | 7:0 | R/W | Link Down Counter. A counter that counts the number of times a link has been down. The counter does not saturate at 255 and is only cleared when writing 0 to the register | 0x00 |

### 7.14.3.11 DEV:PCS1G_CFG_STATUS:PCS1G_STICKY

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G status register for sticky bits

*Table 572 •* **Fields in PCS1G_STICKY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_DOWN_STICKY | 4 | Sticky | The sticky bit is set when the link has been down - i.e. if the ANEG state machine has not been in the AN_DISABLE_LINK_OK or LINK_OK state for one or more clock cycles. This occurs if e.g. ANEG is restarted or for example if signal-detect or synchronization has been lost for more than 10 ms (1.6 ms in SGMII mode). By setting the UDLT bit, the required down time can be reduced to 9,77 us (1.56 us)<br>0: Link is up<br>1: Link has been down<br>Bit is cleared by writing a 1 to this position. | 0x0 |

*Table 572 •* **Fields in PCS1G_STICKY *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OUT_OF_SYNC_STICKY | 0 | Sticky | Sticky bit indicating if PCS synchronization has been lost<br>0: Synchronization has not been lost at any time<br>1: Synchronization has been lost for one or more clock cycles<br>Bit is cleared by writing a 1 to this position. | 0x0 |

### 7.14.3.12 DEV:PCS1G_CFG_STATUS:PCS1G_LPI_CFG

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

Configuration register for Low Power Idle (Energy Efficient Ethernet)

*Table 573 •* **Fields in PCS1G_LPI_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| QSGMII_MS_SEL | 20 | R/W | QSGMII master/slave selection (only one master allowed per QSGMII). The master drives LPI timing on serdes<br>0: Slave<br>1: Master | 0x1 |
| TX_ASSERT_LPIDLE | 0 | R/W | Assert Low-Power Idle (LPI) in transmit mode<br>0: Disable LPI transmission<br>1: Enable LPI transmission | 0x0 |

### 7.14.3.13 DEV:PCS1G_CFG_STATUS:PCS1G_LPI_WAKE_ERROR_CNT

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

PCS1G Low Power Idle wake error counter (Energy Efficient Ethernet)

*Table 574 •* **Fields in PCS1G_LPI_WAKE_ERROR_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WAKE_ERROR_CNT | 15:0 | R/W | Wake Error Counter. A counter that is incremented when the link partner does not send wake-up burst in due time. The counter saturates at 65535 and is cleared when writing 0 to the register | 0x0000 |

### 7.14.3.14 DEV:PCS1G_CFG_STATUS:PCS1G_LPI_STATUS

**Parent:** DEV:PCS1G_CFG_STATUS

**Instances:** 1

Status register for Low Power Idle (Energy Efficient Ethernet)

***Table 575 •*** **Fields in PCS1G_LPI_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_LPI_EVENT_STICKY | 12 | Sticky | Receiver Low-Power idle occurrence<br>0: No LPI symbols received<br>1: Receiver has received LPI symbols<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| RX_QUIET | 9 | R/O | Receiver Low-Power Quiet mode<br>0: Receiver not in quiet mode<br>1: Receiver is in quiet mode | 0x0 |
| RX_LPI_MODE | 8 | R/O | Receiver Low-Power Idle mode<br>0: Receiver not in low power idle mode<br>1: Receiver is in low power idle mode | 0x0 |
| TX_LPI_EVENT_STICKY | 4 | Sticky | Transmitter Low-Power idle occurrence<br>0: No LPI symbols transmitted<br>1: Transmitter has transmitted LPI symbols<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| TX_QUIET | 1 | R/O | Transmitter Low-Power Quiet mode<br>0: Transmitter not in quiet mode<br>1: Transmitter is in quiet mode | 0x0 |
| TX_LPI_MODE | 0 | R/O | Transmitter Low-Power Idle mode<br>0: Transmitter not in low power idle mode<br>1: Transmitter is in low power idle mode | 0x0 |

## 7.14.4 DEV:PCS1G_TSTPAT_CFG_STATUS

**Parent:** DEV

**Instances:** 1

PCS1G testpattern configuration and status register set

***Table 576 •*** **Registers in PCS1G_TSTPAT_CFG_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PCS1G_TSTPAT_MODE_CFG | 0x00000000 | 1 | PCS1G TSTPAT MODE CFG | Page 439 |

*Table 576 •* **Registers in PCS1G_TSTPAT_CFG_STATUS *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PCS1G_TSTPAT_STATUS | 0x00000004 | 1 | PCS1G TSTPAT STATUS | Page 439 |

### 7.14.4.1 DEV:PCS1G_TSTPAT_CFG_STATUS:PCS1G_TSTPAT_MODE_CFG

**Parent:** DEV:PCS1G_TSTPAT_CFG_STATUS

**Instances:** 1

PCS1G testpattern mode configuration register (Frame based pattern 4 and 5 might be not available depending on chip type)

*Table 577 •* **Fields in PCS1G_TSTPAT_MODE_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| JTP_SEL | 2:0 | R/W | Jitter Test Pattern Select: Enables and selects the jitter test pattern to be transmitted. The jitter test patterns are according to the IEEE 802.3, Annex 36A<br>0: Disable transmission of test patterns<br>1: High frequency test pattern - repeated transmission of D21.5 code group<br>2: Low frequency test pattern - repeated transmission of K28.7 code group<br>3: Mixed frequency test pattern - repeated transmission of K28.5 code group<br>4: Long continuous random test pattern (packet length is 1524 bytes)<br>5: Short continuous random test pattern (packet length is 360 bytes) | 0x0 |

### 7.14.4.2 DEV:PCS1G_TSTPAT_CFG_STATUS:PCS1G_TSTPAT_STATUS

**Parent:** DEV:PCS1G_TSTPAT_CFG_STATUS

**Instances:** 1

PCS1G testpattern status register

*Table 578 •* **Fields in PCS1G_TSTPAT_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| JTP_ERR_CNT | 15:8 | R/W | Jitter Test Pattern Error Counter. The counter saturates at 255 and is cleared by writing 0 to the register. | 0x00 |
| JTP_ERR | 4 | R/O | Jitter Test Pattern Error<br>0: Jitter pattern checker has found no error<br>1: Jitter pattern checker has found an error | 0x0 |
| JTP_LOCK | 0 | R/O | Jitter Test Pattern Lock<br>0: Jitter pattern checker has not locked<br>1: Jitter pattern checker has locked | 0x0 |

## 7.14.5 DEV:PCS_FX100_CONFIGURATION

**Parent:** DEV

**Instances:** 1

Configuration register set for PCS 100Base-FX logic

*Table 579 •* **Registers in PCS_FX100_CONFIGURATION**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PCS_FX100_CFG | 0x00000000 | 1 | PCS 100Base FX Configuration | Page 440 |

### 7.14.5.1 DEV:PCS_FX100_CONFIGURATION:PCS_FX100_CFG

**Parent:** DEV:PCS_FX100_CONFIGURATION

**Instances:** 1

Configuration bit groups for 100Base-FX PCS

*Table 580 •* **Fields in PCS_FX100_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SD_SEL | 26 | R/W | Signal detect selection (select input for internal signal_detect line)<br>0: Select signal_detect line from hardmacro<br>1: Select external signal_detect line | 0x0 |
| RESERVED | 25 | R/W | Must be set to its default. | 0x1 |

*Table 580 •* **Fields in PCS_FX100_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SD_ENA | 24 | R/W | Signal Detect Enable<br>0: The Signal Detect input pin is ignored. The PCS assumes an active Signal Detect at all times<br>1: The Signal Detect input pin is used to determine if a signal is detected | 0x1 |
| RESERVED | 15:12 | R/W | Must be set to its default. | 0x4 |
| LINKHYSTTIMER | 7:4 | R/W | Link hysteresis timer configuration. The hysteresis time lasts [linkhysttimer] * 65536 ns + 2320 ns. If linkhysttime is set to 5, the hysteresis lasts the minimum time of 330 us as specified in IEEE 802.3 - 24.3.3.4. | 0x5 |
| UNIDIR_MODE_ENA | 3 | R/W | Unidirectional mode enable. Implementation 0f 802.3 clause 66. When asserted, this enables MAC to transmit data independent of the state of the receive link.<br>0: Unidirectional mode disabled<br>1: Unidirectional mode enabled | 0x0 |
| FEFCHK_ENA | 2 | R/W | Far-End Fault (FEF) detection enable<br>0: Disable FEF detection<br>1 Enable FEF detection | 0x1 |
| FEFGEN_ENA | 1 | R/W | Far-End Fault (FEF) generation enable<br>0: Disable FEF generation<br>1 Enable FEF generation | 0x1 |
| PCS_ENA | 0 | R/W | PCS enable<br>0: Disable PCS<br>1: Enable PCS | 0x0 |

## 7.14.6 DEV:PCS_FX100_STATUS

**Parent:** DEV

**Instances:** 1

Status register set for PCS 100Base-FX logic

*Table 581 •* **Registers in PCS_FX100_STATUS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PCS_FX100_STATUS | 0x00000000 | 1 | PCS 100Base FX Status | Page 441 |

### 7.14.6.1 DEV:PCS_FX100_STATUS:PCS_FX100_STATUS

**Parent:** DEV:PCS_FX100_STATUS

**Instances:** 1

Status bit groups for 100Base-FX PCS. Note: If sigdet_cfg != "00" is selected status signal "signal_detect" shows the internal signal_detect value is gated with the status of rx toggle-rate control circuitry.

***Table 582 •*** **Fields in PCS_FX100_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PCS_ERROR_STICKY | 7 | Sticky | PCS error has occurred<br>1: RX_ER was high while RX_DV active<br>0: No RX_ER indication found while RX_DV active<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| FEF_FOUND_STICKY | 6 | Sticky | Far-end Fault state has occurred<br>1: A Far-End Fault has been detected<br>0: No Far-End Fault occurred<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| SSD_ERROR_STICKY | 5 | Sticky | Stream Start Delimiter error occurred<br>1: A Start-of-Stream Delimiter error has been detected<br>0: No SSD error occurred<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| SYNC_LOST_STICKY | 4 | Sticky | Synchronization lost<br>1: Synchronization lost<br>0: No sync lost occurred<br>Bit is cleared by writing a 1 to this position. | 0x0 |
| FEF_STATUS | 2 | R/O | Current status of Far-end Fault detection state<br>1: Link currently in fault state<br>0: Link is in normal state | 0x0 |
| SIGNAL_DETECT | 1 | R/O | Current status of selected signal_detect input line<br>1: Proper signal detected<br>0: No proper signal found | 0x0 |
| SYNC_STATUS | 0 | R/O | Status of synchronization<br>1: Link established<br>0: No link found | 0x0 |

# 7.15 ICPU_CFG

*Table 583* • **Register Groups in ICPU_CFG**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CPU_SYSTEM_CTRL | 0x00000000 | 1 | Configurations for the CPU system. | Page 443 |
| PI_MST | 0x0000002C | 1 | Parallel Interface Master Configuration | Page 446 |
| SPI_MST | 0x00000050 | 1 | SPI Master Configuration | Page 449 |
| INTR | 0x00000084 | 1 | Interrupt Registers | Page 451 |
| GPDMA | 0x0000013C | 1 | Frame DMA | Page 484 |
| INJ_FRM_SPC | 0x00000188 | 8<br>0x00000010 | Injection frame spacing | Page 489 |
| TIMERS | 0x00000208 | 1 | Timer Registers | Page 490 |
| MEMCTRL | 0x00000234 | 1 | DDR2/3 Memory Controller Registers | Page 494 |
| TWI_DELAY | 0x000002A4 | 1 | Configuration registers | Page 504 |

## 7.15.1 ICPU_CFG:CPU_SYSTEM_CTRL

**Parent:** ICPU_CFG

**Instances:** 1

*Table 584* • **Registers in CPU_SYSTEM_CTRL**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| GPR | 0x00000000 | 8<br>0x00000004 | General Purpose Register | Page 443 |
| RESET | 0x00000020 | 1 | Reset Settings | Page 444 |
| GENERAL_CTRL | 0x00000024 | 1 | General control | Page 445 |
| GENERAL_STAT | 0x00000028 | 1 | General status | Page 446 |

### 7.15.1.1 ICPU_CFG:CPU_SYSTEM_CTRL:GPR

**Parent:** ICPU_CFG:CPU_SYSTEM_CTRL

**Instances:** 8

*Table 585 •* **Fields in GPR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GPR | 31:0 | R/W | General purpose 8 times 32-bit registers for software development and debug. | 0x00000000 |

### 7.15.1.2 ICPU_CFG:CPU_SYSTEM_CTRL:RESET

**Parent:** ICPU_CFG:CPU_SYSTEM_CTRL

**Instances:** 1

*Table 586 •* **Fields in RESET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CPU_RELEASE | 4 | R/W | Set this field to enable the VCore CPU. This field is only valid when automatic booting of the VCore CPU has been disabled via VCore_Cfg inputs. This field has no effect when the VCore CPU is configured for automatically boot.<br>Note: By using this field it is possible for an external CPU to manually load a code image to memory, change into normal mode, and then release the VCore CPU after which it will boot from memory rather than FLASH.<br>0: VCore CPU is forced in reset<br>1: VCore CPU is allowed to boot | 0x0 |
| CORE_RST_CPU_ONLY | 3 | R/W | Set this field to enable VCore System reset protection. It is possible to protect the VCore System from soft-reset (issued via RESET:CORE_RST_FORCE) and watchdog-timeout. When this field is set the aforementioned resets only reset the VCore CPU, not the VCore System.<br>0: WDT event reset entire VCore<br>1: WDT event only reset the VCore CPU | 0x0 |

*Table 586 •* **Fields in RESET *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CORE_RST_PROTECT | 2 | R/W | Set this field to enable VCore reset protection. It is possible to protect the entire VCore from chip-level soft-reset (issued via DEVCPU_GCB::SOFT_CHIP_RST.SOFT_CHIP_RST). Setting this field does not protect against hard-reset of the chip (by asserting the reset pin).<br>0: No reset protection<br>1: VCore is protected from chip-level-soft-reset | 0x0 |
| CORE_RST_FORCE | 1 | One-shot | Set this field to generate a soft reset for the VCore. This field will be cleared when the reset has taken effect.<br>It is possible to protect the VCore system (everything else than the VCore CPU) from reset via RESET.CORE_RST_CPU_ONLY.<br>0: VCore is not reset<br>1: Initiate soft reset of the VCore | 0x0 |
| MEM_RST_FORCE | 0 | R/W | While this field is set, the memory controller is held in reset.<br>0: Memory controller is not reset<br>1: Memory controller is forced in reset | 0x1 |

### 7.15.1.3 ICPU_CFG:CPU_SYSTEM_CTRL:GENERAL_CTRL

**Parent:** ICPU_CFG:CPU_SYSTEM_CTRL

**Instances:** 1

*Table 587 •* **Fields in GENERAL_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IF_MASTER_PI_ENA | 1 | R/W | Set this field to force PI interface into master mode. By default only the boot interface of the VCore system is in master mode (controlled by the VCore).<br>This field must be set if the VCore is started manually or requires the non-boot interface for accessing logic outside the chip. Please note, if this field is set, it is no longer possible for an external CPU to access registers in the chip via PI. | 0x0 |

*Table 587 •* **Fields in GENERAL_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BOOT_MODE_ENA | 0 | R/W | Use this field to change from Boot mode to Normal mode. In Boot mode, the reset vector of the VCore CPU maps to CS0 on the parallel interface. When in Normal mode, this address maps instead to the DRAM Controller. The DRAM Controller must be operational before disabling Boot mode.<br>After setting Boot mode, this register must be read back. The change in Boot mode becomes effective during reading.<br>0: The VCore memory map is in Normal mode.<br>1: The VCore memory map is in Boot mode. | 0x1 |

#### 7.15.1.4 ICPU_CFG:CPU_SYSTEM_CTRL:GENERAL_STAT

**Parent:** ICPU_CFG:CPU_SYSTEM_CTRL

**Instances:** 1

*Table 588 •* **Fields in GENERAL_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CPU_SLEEP | 3 | R/O | This field is set if the VCore CPU has entered sleep mode. | 0x0 |
| ENDIAN_MODE | 2 | R/O | This field shows the endianess that has been configured for the VCore CPU.<br>0: Little Endian<br>1: Big Endian | 0x0 |
| BOOT_MODE | 1 | R/O | This field shows which boot strategy that has been configured for the VCore CPU.<br>0: Automatic booting<br>1: Manual booting | 0x0 |
| BOOT_IF | 0 | R/O | This field shows which boot interface that has been configured for the VCore CPU.<br>0: PI<br>1: SPI | 0x0 |

### 7.15.2 ICPU_CFG:PI_MST

**Parent:** ICPU_CFG

**Instances:** 1

*Table 589 •* **Registers in PI_MST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PI_MST_CFG | 0x00000000 | 1 | PI Master Configuration | Page 447 |
| PI_MST_CTRL | 0x00000004 | 4<br>0x00000004 | PI Master Control Register | Page 447 |
| PI_MST_STATUS | 0x00000014 | 4<br>0x00000004 | PI Master Status Registers | Page 449 |

### 7.15.2.1 ICPU_CFG:PI_MST:PI_MST_CFG

**Parent:** ICPU_CFG:PI_MST

**Instances:** 1

*Table 590 •* **Fields in PI_MST_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 5 | R/W | Must be set to its default. | 0x1 |
| CLK_DIV | 4:0 | R/W | Controls the clock for the PI Controller.<br>0: Illegal<br>1: Illegal<br>2: Use CPU clock/2<br>...<br>31: Use CPU clock/31 | 0x1F |

### 7.15.2.2 ICPU_CFG:PI_MST:PI_MST_CTRL

**Parent:** ICPU_CFG:PI_MST

**Instances:** 4

This is a replicated register, where each replication holds the configurations for one chip select. Changes to a value in one of the replicated instances apply only to that chip select.

*Table 591 •* **Fields in PI_MST_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DATA_WID | 23 | R/W | Data width. In 8-bit mode, the unused data-bits contain additional address information.<br>0: 8 bits<br>1: 16 bits | 0x0 |
| DEVICE_PACED_XFER_ENA | 22 | R/W | Device-paced transfer enable. When enabled, use PI_nDone to end a transfer.<br>0: Disabled<br>1: Enabled | 0x0 |

*Table 591 •* **Fields in PI_MST_CTRL** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DEVICE_PACED_TIMEOUT_ENA | 21 | R/W | Enable timeout on device-paced transfers. If enabled, a device_paced_transfer transfer does not wait indefinitely for assertion of PI_nDone. If a timeout occurs, the TIMEOUT_ERR_STICKY bit is set in the status register and the current transfer is terminated (read-data will be invalid).<br>When enabling device paced timeout ICPU_CFG::PI_MST_CTRL.CSCC field must be set higher than 0 and the timeout defined by ICPU_CFG::PI_MST_CTRL.DEVICE_PACED_TIMEOUT must be higher than ICPU_CFG::PI_MST_CTRL.WAITCC. | 0x0 |
| DEVICE_PACED_TIMEOUT | 20:18 | R/W | Determines the number of PI_Clk cycles from the start of a transfer until a timeout occurs. This field is only valid when timeout for device-paced transfer is enabled.<br>000: 16 PI_Clk cycles<br>001: 32 PI_Clk cycles<br>010: 64 PI_Clk cycles<br>011: 128 PI_Clk cycles<br>100: 256 PI_Clk cycles<br>101: 512 PI_Clk cycles<br>110: 1024 PI_Clk cycles<br>111: 2048 PI_Clk cycles | 0x0 |
| RESERVED | 17 | R/W | Must be set to its default. | 0x1 |
| DONE_POL | 16 | R/W | Polarity of PI_nDone for device-paced transfers.<br>0: PI_nDone is active low<br>1: PI_nDone is active high | 0x0 |
| SMPL_ON_DONE | 15 | R/W | Controls when data is sampled in relation to assertion of PI_nDone for device-paced reads.<br>0: Data is sampled one PI_Clk cycle after PI_nDone goes active.<br>1: Data is sampled on the same PI_Clk cycle where PI_nDone goes active. | 0x0 |
| WAITCC | 14:7 | R/W | Number of wait states measured in PI_Clk cycles on both read and write transfers. | 0x01 |
| CSCC | 6:5 | R/W | Number of PI_Clk cycles from address driven to PI_nCS[x] low. | 0x1 |
| OECC | 4:3 | R/W | Number of PI_Clk cycles from PI_nCS[x] low to PI_nOE low. | 0x0 |
| HLDCC | 2:0 | R/W | Number of PI_Clk cycles to insert at the end of a transfer. | 0x0 |

### 7.15.2.3 ICPU_CFG:PI_MST:PI_MST_STATUS

**Parent:** ICPU_CFG:PI_MST

**Instances:** 4

This is a replicated register, where each replication holds the status for one chip select.

***Table 592 •*** **Fields in PI_MST_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMEOUT_ERR_STICKY | 0 | Sticky | If a timeout is enabled and timeout occurs during a device-paced transfer, this bit is set. | 0x0 |

## 7.15.3 ICPU_CFG:SPI_MST

**Parent:** ICPU_CFG

**Instances:** 1

***Table 593 •*** **Registers in SPI_MST**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SPI_MST_CFG | 0x00000000 | 1 | SPI Master Configuration | Page 449 |
| SW_MODE | 0x00000014 | 1 | Manual control of the SPI interface | Page 450 |

### 7.15.3.1 ICPU_CFG:SPI_MST:SPI_MST_CFG

**Parent:** ICPU_CFG:SPI_MST

**Instances:** 1

***Table 594 •*** **Fields in SPI_MST_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FAST_READ_ENA | 10 | R/W | The type of read-instruction that the SPI Controller generates for reads.<br>0: READ (slow read - Instruction code - 0x03)<br>1: FAST READ (fast read - Instruction code - 0x0B) | 0x0 |
| CS_DESELECT_TIME | 9:5 | R/W | The minimum number of SPI clock cycles for which the SPI chip select (SI_nEn) must be deasserted in between transfers. Typical value of this is 100 ns. Setting this field to 0 is illegal. | 0x1F |

*Table 594 •* **Fields in SPI_MST_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLK_DIV | 4:0 | R/W | Controls the clock frequency for the SPI interface (SI_Clk). The clock frequency is VCore system clock divided by the value of this field. Setting this field to 0 or 1 value is illegal. | 0x1F |

### 7.15.3.2 ICPU_CFG:SPI_MST:SW_MODE

**Parent:** ICPU_CFG:SPI_MST

**Instances:** 1

Note: There are 4 chip selects in total, but only chip select 0 is mapped to IO-pin (SI_nEn). The rest of the SPI chip selects are available as alternate functions on GPIOs, these must be enabled in the GPIO controller before they can be controlled via this register.

*Table 595 •* **Fields in SW_MODE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW_PIN_CTRL_MODE | 13 | R/W | Set to enable software pin control mode (Bit banging), when set software has direct control of the SPI interface.<br>This mode is used for writing into flash. | 0x0 |
| SW_SPI_SCK | 12 | R/W | Value to drive on SI_Clk output. This field is only used if SW_MODE.SW_PIN_CTRL_MODE is set. | 0x0 |
| SW_SPI_SCK_OE | 11 | R/W | Set to enable drive of SI_Clk output. This field is only used if SW_MODE.SW_PIN_CTRL_MODE is set. | 0x0 |
| SW_SPI_SDO | 10 | R/W | Value to drive on SI_DO output. This field is only used if SW_MODE.SW_PIN_CTRL_MODE is set. | 0x0 |
| SW_SPI_SDO_OE | 9 | R/W | Set to enable drive of SI_DO output. This field is only used if SW_MODE.SW_PIN_CTRL_MODE is set. | 0x0 |
| SW_SPI_CS | 8:5 | R/W | Value to drive on SI_nEn outputs, each bit in this field maps to a corresponding chip-select (0 though 3). This field is only used if SW_MODE.SW_PIN_CTRL_MODE is set.<br>Note: Chip selects 1 though 3 are available as alternate GPIO functions. | 0x0 |

*Table 595 •* **Fields in SW_MODE *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW_SPI_CS_OE | 4:1 | R/W | Set to enable drive of SI_nEn outputs, each bit in this field maps to a corresponding chip-select (0 though 3). This field is only used if SW_MODE.SW_PIN_CTRL_MODE is set.<br>Note: Chip selects 1 though 3 are available as alternate GPIO functions. | 0x0 |
| SW_SPI_SDI | 0 | R/O | Current value of the SI_DI input. | 0x0 |

## 7.15.4 ICPU_CFG:INTR

**Parent:** ICPU_CFG

**Instances:** 1

*Table 596 •* **Registers in INTR**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| INTR | 0x00000000 | 1 | Interrupt sticky bits | Page 453 |
| INTR_ENA | 0x00000004 | 1 | Interrupt enable | Page 456 |
| INTR_ENA_CLR | 0x00000008 | 1 | Clear interrupt enable | Page 457 |
| INTR_ENA_SET | 0x0000000C | 1 | Set interrupt enable | Page 458 |
| INTR_RAW | 0x00000010 | 1 | Raw of interrupt source | Page 460 |
| ICPU_IRQ0_ENA | 0x00000014 | 1 | Enable of ICPU_IRQ0 interrupt | Page 461 |
| ICPU_IRQ0_IDENT | 0x00000018 | 1 | Sources of ICPU_IRQ0 interrupt | Page 461 |
| ICPU_IRQ1_ENA | 0x0000001C | 1 | Enable of ICPU_IRQ1 interrupt | Page 463 |
| ICPU_IRQ1_IDENT | 0x00000020 | 1 | Sources of ICPU_IRQ1 interrupt | Page 463 |
| EXT_IRQ0_ENA | 0x00000024 | 1 | Enable of EXT_IRQ0 interrupt | Page 464 |
| EXT_IRQ0_IDENT | 0x00000028 | 1 | Sources of EXT_IRQ0 interrupt | Page 465 |
| EXT_IRQ1_ENA | 0x0000002C | 1 | Enable of EXT_IRQ1 interrupt | Page 466 |
| EXT_IRQ1_IDENT | 0x00000030 | 1 | Sources of EXT_IRQ1 interrupt | Page 466 |
| DEV_IDENT | 0x00000034 | 1 | Device interrupts | Page 467 |
| EXT_IRQ0_INTR_CFG | 0x00000038 | 1 | EXT_IRQ0 interrupt configuration | Page 468 |

*Table 596 •* **Registers in INTR *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| EXT_IRQ1_INTR_CFG | 0x0000003C | 1 | EXT_IRQ1 interrupt configuration | Page 469 |
| SW0_INTR_CFG | 0x00000040 | 1 | SW0 interrupt configuration | Page 470 |
| SW1_INTR_CFG | 0x00000044 | 1 | SW1 interrupt configuration | Page 471 |
| MIIM1_INTR_CFG | 0x00000048 | 1 | MIIM1 interrupt configuration | Page 471 |
| MIIM0_INTR_CFG | 0x0000004C | 1 | MIIM0 interrupt configuration | Page 472 |
| PI_SD0_INTR_CFG | 0x00000050 | 1 | PI_SD0 interrupt configuration | Page 473 |
| PI_SD1_INTR_CFG | 0x00000054 | 1 | PI_SD1 interrupt configuration | Page 473 |
| UART_INTR_CFG | 0x00000058 | 1 | UART interrupt configuration | Page 474 |
| TIMER0_INTR_CFG | 0x0000005C | 1 | TIMER0 interrupt configuration | Page 475 |
| TIMER1_INTR_CFG | 0x00000060 | 1 | TIMER1 interrupt configuration | Page 475 |
| TIMER2_INTR_CFG | 0x00000064 | 1 | TIMER2 interrupt configuration | Page 475 |
| FDMA_INTR_CFG | 0x00000068 | 1 | FDMA interrupt configuration | Page 476 |
| TWI_INTR_CFG | 0x0000006C | 1 | TWI interrupt configuration | Page 476 |
| GPIO_INTR_CFG | 0x00000070 | 1 | GPIO interrupt configuration | Page 477 |
| SGPIO_INTR_CFG | 0x00000074 | 1 | SGPIO interrupt configuration | Page 478 |
| DEV_ALL_INTR_CFG | 0x00000078 | 1 | DEV_ALL interrupt configuration | Page 478 |
| BLK_ANA_INTR_CFG | 0x00000078 | 1 | BLK_ANA interrupt configuration | Page 479 |
| XTR_RDY0_INTR_CFG | 0x00000080 | 1 | XTR_RDY0 interrupt configuration | Page 480 |
| XTR_RDY1_INTR_CFG | 0x00000084 | 1 | XTR_RDY1 interrupt configuration | Page 480 |
| INJ_RDY0_INTR_CFG | 0x00000090 | 1 | INJ_RDY0 interrupt configuration | Page 481 |
| INJ_RDY1_INTR_CFG | 0x00000094 | 1 | INJ_RDY1 interrupt configuration | Page 482 |
| INTEGRITY_INTR_CFG | 0x000000A4 | 1 | INTEGRITY interrupt configuration | Page 482 |
| PTP_SYNC_INTR_CFG | 0x000000A8 | 1 | PTP_SYNC interrupt configuration | Page 483 |

*Table 596 •* **Registers in INTR *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| DEV_ENA | 0x000000AC | 1 | Device Interrupt enable | Page 484 |

### 7.15.4.1 ICPU_CFG:INTR:INTR

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Asserted for the active interrupt sources.

*Table 597 •* **Fields in INTR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM1_INTR | 28 | Sticky | This field is set when MIIM master1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the MIIM master1 interrupt event is no longer active. | 0x0 |
| MIIM0_INTR | 27 | Sticky | This field is set when MIIM master0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the MIIM master0 interrupt event is no longer active. | 0x0 |
| PTP_SYNC_INTR | 26 | Sticky | This field is set when PTP-Sync interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the PTP-Sync interrupt event is no longer active. | 0x0 |
| INTEGRITY_INTR | 25 | Sticky | This field is set when integrity interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if there are no longer any pending integrity interrupt event. | 0x0 |
| INJ_RDY1_INTR | 21 | Sticky | This field is set when inj-group-1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the inj-group-1 interrupt event is no longer active. | 0x0 |
| INJ_RDY0_INTR | 20 | Sticky | This field is set when inj-group-0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the inj-group-0 interrupt event is no longer active. | 0x0 |

*Table 597 •* **Fields in INTR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| XTR_RDY1_INTR | 17 | Sticky | This field is set when xtr-group-1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the xtr-group-1 interrupt event is no longer active. | 0x0 |
| XTR_RDY0_INTR | 16 | Sticky | This field is set when xtr-group-0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the xtr-group-0 interrupt event is no longer active. | 0x0 |
| BLK_ANA_INTR | 15 | Sticky | This field is set when analyzer interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the analyzer interrupt event is no longer active. | 0x0 |
| DEV_ALL_INTR | 14 | Sticky | This field is set when interrupt from any device (port) is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if there is still a pending interrupt from any device. This is a cascaded interrupt, read DEV_IDENT to see which device(s) that is/are currently interrupting. | 0x0 |
| SGPIO_INTR | 13 | Sticky | This field is set when Serial-GPIO interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the Serial-GPIO interrupt event is no longer active. | 0x0 |
| GPIO_INTR | 12 | Sticky | This field is set when Parallel-GPIO interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the Parallel-GPIO interrupt event is no longer active. | 0x0 |
| TWI_INTR | 11 | Sticky | This field is set when TWI interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the TWI interrupt event is no longer active. | 0x0 |
| FDMA_INTR | 10 | Sticky | This field is set when FDMA interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the FDMA interrupt event is no longer active. | 0x0 |

*Table 597 •* **Fields in INTR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER2_INTR | 9 | Sticky | This field is set when Timer-2 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the Timer-2 interrupt event is no longer active. | 0x0 |
| TIMER1_INTR | 8 | Sticky | This field is set when Timer-1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the Timer-1 interrupt event is no longer active. | 0x0 |
| TIMER0_INTR | 7 | Sticky | This field is set when Timer-0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the Timer-0 interrupt event is no longer active. | 0x0 |
| UART_INTR | 6 | Sticky | This field is set when UART interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the UART interrupt event is no longer active. | 0x0 |
| PI_SD1_INTR | 5 | Sticky | This field is set when PI-Slow-Done-1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the PI-Slow-Done-1 interrupt event is no longer active. | 0x0 |
| PI_SD0_INTR | 4 | Sticky | This field is set when PI-Slow-Done-0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the PI-Slow-Done-0 interrupt event is no longer active. | 0x0 |
| SW1_INTR | 3 | Sticky | This field is set when SW1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the SW1 interrupt event is no longer active. | 0x0 |
| SW0_INTR | 2 | Sticky | This field is set when SW0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the SW0 interrupt event is no longer active. | 0x0 |
| EXT_IRQ1_INTR | 1 | Sticky | This field is set when EXT_IRQ1 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the EXT_IRQ1 interrupt event is no longer active. | 0x0 |

*Table 597 •* **Fields in INTR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_INTR | 0 | Sticky | This field is set when EXT_IRQ0 interrupt is detected. Clearing of this field is done by writing 1 to this field. This field can only be cleared if the EXT_IRQ0 interrupt event is no longer active. | 0x0 |

### 7.15.4.2 ICPU_CFG:INTR:INTR_ENA

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Controls if active interrupt indications (from INTR) can propagate to their destinations. In a multi-threaded environment, or with more than one active processor the INTR_ENA_SET and INTR_ENA_CLR registers can be used for atomic modifications of this register. Writing 1 to any bit(s) in the INTR_ENA_SET register will set the corresponding bit(s) in this register, Writing 1 to any bit in the INTR_ENA_CLR register will clear the corresponding bit(s) in this register.

*Table 598 •* **Fields in INTR_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM1_INTR_ENA | 28 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| MIIM0_INTR_ENA | 27 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| PTP_SYNC_INTR_ENA | 26 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| INTEGRITY_INTR_ENA | 25 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| INJ_RDY1_INTR_ENA | 21 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| INJ_RDY0_INTR_ENA | 20 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| XTR_RDY1_INTR_ENA | 17 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| XTR_RDY0_INTR_ENA | 16 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| BLK_ANA_INTR_ENA | 15 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| DEV_ALL_INTR_ENA | 14 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| SGPIO_INTR_ENA | 13 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| GPIO_INTR_ENA | 12 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| TWI_INTR_ENA | 11 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |

*Table 598 •* **Fields in INTR_ENA *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FDMA_INTR_ENA | 10 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| TIMER2_INTR_ENA | 9 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| TIMER1_INTR_ENA | 8 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| TIMER0_INTR_ENA | 7 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| UART_INTR_ENA | 6 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| PI_SD1_INTR_ENA | 5 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| PI_SD0_INTR_ENA | 4 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| SW1_INTR_ENA | 3 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| SW0_INTR_ENA | 2 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| EXT_IRQ1_INTR_ENA | 1 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |
| EXT_IRQ0_INTR_ENA | 0 | R/W | Set this field to enable the interrupt to propagate to its destination. | 0x0 |

### 7.15.4.3 ICPU_CFG:INTR:INTR_ENA_CLR

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 599 •* **Fields in INTR_ENA_CLR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM1_INTR_ENA_CLR | 28 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| MIIM0_INTR_ENA_CLR | 27 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| PTP_SYNC_INTR_ENA_CLR | 26 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| INTEGRITY_INTR_ENA_CLR | 25 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| INJ_RDY1_INTR_ENA_CLR | 21 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| INJ_RDY0_INTR_ENA_CLR | 20 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| XTR_RDY1_INTR_ENA_CLR | 17 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |

*Table 599 •* **Fields in INTR_ENA_CLR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| XTR_RDY0_INTR_ENA_CLR | 16 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| BLK_ANA_INTR_ENA_CLR | 15 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| DEV_ALL_INTR_ENA_CLR | 14 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| SGPIO_INTR_ENA_CLR | 13 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| GPIO_INTR_ENA_CLR | 12 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| TWI_INTR_ENA_CLR | 11 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| FDMA_INTR_ENA_CLR | 10 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| TIMER2_INTR_ENA_CLR | 9 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| TIMER1_INTR_ENA_CLR | 8 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| TIMER0_INTR_ENA_CLR | 7 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| UART_INTR_ENA_CLR | 6 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| PI_SD1_INTR_ENA_CLR | 5 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| PI_SD0_INTR_ENA_CLR | 4 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| SW1_INTR_ENA_CLR | 3 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| SW0_INTR_ENA_CLR | 2 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| EXT_IRQ1_INTR_ENA_CLR | 1 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |
| EXT_IRQ0_INTR_ENA_CLR | 0 | One-shot | Set to clear corresponding interrupt enable in INTR_ENA. | 0x0 |

### 7.15.4.4 ICPU_CFG:INTR:INTR_ENA_SET

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 600 •* **Fields in INTR_ENA_SET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM1_INTR_ENA_SET | 28 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |

*Table 600 •* **Fields in INTR_ENA_SET *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM0_INTR_ENA_SET | 27 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| PTP_SYNC_INTR_ENA_SET | 26 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| INTEGRITY_INTR_ENA_SET | 25 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| INJ_RDY1_INTR_ENA_SET | 21 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| INJ_RDY0_INTR_ENA_SET | 20 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| XTR_RDY1_INTR_ENA_SET | 17 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| XTR_RDY0_INTR_ENA_SET | 16 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| BLK_ANA_INTR_ENA_SET | 15 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| DEV_ALL_INTR_ENA_SET | 14 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| SGPIO_INTR_ENA_SET | 13 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| GPIO_INTR_ENA_SET | 12 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| TWI_INTR_ENA_SET | 11 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| FDMA_INTR_ENA_SET | 10 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| TIMER2_INTR_ENA_SET | 9 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| TIMER1_INTR_ENA_SET | 8 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| TIMER0_INTR_ENA_SET | 7 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| UART_INTR_ENA_SET | 6 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| PI_SD1_INTR_ENA_SET | 5 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| PI_SD0_INTR_ENA_SET | 4 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| SW1_INTR_ENA_SET | 3 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| SW0_INTR_ENA_SET | 2 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |
| EXT_IRQ1_INTR_ENA_SET | 1 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |

*Table 600 •* **Fields in INTR_ENA_SET *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_INTR_ENA_SET | 0 | One-shot | Set this field to set corresponding interrupt enable in INTR_ENA. | 0x0 |

### 7.15.4.5 ICPU_CFG:INTR:INTR_RAW

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Shows the current value of the interrupt source to the interrupt controller (interrupts are active high). External interrupt inputs are corrected for polarity before being presented in this register.

*Table 601 •* **Fields in INTR_RAW**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM1_RAW | 28 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| MIIM0_RAW | 27 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| PTP_SYNC_RAW | 26 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| INTEGRITY_RAW | 25 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| INJ_RDY1_RAW | 21 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| INJ_RDY0_RAW | 20 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| XTR_RDY1_RAW | 17 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| XTR_RDY0_RAW | 16 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| BLK_ANA_RAW | 15 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| DEV_ALL_RAW | 14 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| SGPIO_RAW | 13 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| GPIO_RAW | 12 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| TWI_RAW | 11 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| FDMA_RAW | 10 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| TIMER2_RAW | 9 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| TIMER1_RAW | 8 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |

*Table 601 •* **Fields in INTR_RAW *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER0_RAW | 7 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| UART_RAW | 6 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| PI_SD1_RAW | 5 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| PI_SD0_RAW | 4 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| SW1_RAW | 3 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| SW0_RAW | 2 | R/O | Current value of interrupt source input to the interrupt controller. | 0x0 |
| EXT_IRQ1_RAW | 1 | R/O | Current value of interrupt source input to the interrupt controller, is already corrected for polarity as configured via EXT_IRQ1_INTR_CFG.EXT_IRQ1_INTR_POL. | 0x0 |
| EXT_IRQ0_RAW | 0 | R/O | Current value of interrupt source input to the interrupt controller, is already corrected for polarity as configured via EXT_IRQ0_INTR_CFG.EXT_IRQ0_INTR_POL. | 0x0 |

### 7.15.4.6 ICPU_CFG:INTR:ICPU_IRQ0_ENA

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 602 •* **Fields in ICPU_IRQ0_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ICPU_IRQ0_ENA | 0 | R/W | Enables ICPU_IRQ0 interrupt<br>0: Interrupt is disabled<br>1: Interrupt is enabled | 0x0 |

### 7.15.4.7 ICPU_CFG:INTR:ICPU_IRQ0_IDENT

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Identifies the source(s) of an active interrupt on output interrupt: ICPU_IRQ0. All asserted interrupts are shown as active high.

*Table 603* • **Fields in ICPU_IRQ0_IDENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ICPU_IRQ0_MIIM1_IDENT | 28 | R/O | Set when MIIM1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_MIIM0_IDENT | 27 | R/O | Set when MIIM0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_PTP_SYNC_IDENT | 26 | R/O | Set when PTP_SYNC interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_INTEGRITY_IDENT | 25 | R/O | Set when INTEGRITY interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_INJ_RDY1_IDENT | 21 | R/O | Set when INJ_RDY1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_INJ_RDY0_IDENT | 20 | R/O | Set when INJ_RDY0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_XTR_RDY1_IDENT | 17 | R/O | Set when XTR_RDY1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_XTR_RDY0_IDENT | 16 | R/O | Set when XTR_RDY0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_BLK_ANA_IDENT | 15 | R/O | Set when BLK_ANA interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_DEV_ALL_IDENT | 14 | R/O | Set when DEV_ALL interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_SGPIO_IDENT | 13 | R/O | Set when SGPIO interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_GPIO_IDENT | 12 | R/O | Set when GPIO interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_TWI_IDENT | 11 | R/O | Set when TWI interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_FDMA_IDENT | 10 | R/O | Set when FDMA interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_TIMER2_IDENT | 9 | R/O | Set when TIMER2 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_TIMER1_IDENT | 8 | R/O | Set when TIMER1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_TIMER0_IDENT | 7 | R/O | Set when TIMER0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_UART_IDENT | 6 | R/O | Set when UART interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_PI_SD1_IDENT | 5 | R/O | Set when PI_SD1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_PI_SD0_IDENT | 4 | R/O | Set when PI_SD0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_SW1_IDENT | 3 | R/O | Set when SW1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |

*Table 603 •* **Fields in ICPU_IRQ0_IDENT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ICPU_IRQ0_SW0_IDENT | 2 | R/O | Set when SW0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_EXT_IRQ1_IDENT | 1 | R/O | Set when EXT_IRQ1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ0_EXT_IRQ0_IDENT | 0 | R/O | Set when EXT_IRQ0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |

### 7.15.4.8 ICPU_CFG:INTR:ICPU_IRQ1_ENA

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 604 •* **Fields in ICPU_IRQ1_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ICPU_IRQ1_ENA | 0 | R/W | Enables ICPU_IRQ1 interrupt<br>0: Interrupt is disabled<br>1: Interrupt is enabled | 0x0 |

### 7.15.4.9 ICPU_CFG:INTR:ICPU_IRQ1_IDENT

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Identifies the source(s) of an active interrupt on output interrupt: ICPU_IRQ1. All asserted interrupts are shown as active high.

*Table 605 •* **Fields in ICPU_IRQ1_IDENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ICPU_IRQ1_MIIM1_IDENT | 28 | R/O | Set when MIIM1 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ1_MIIM0_IDENT | 27 | R/O | Set when MIIM0 interrupt is a source of the ICPU_IRQ0 interrupt. | 0x0 |
| ICPU_IRQ1_PTP_SYNC_IDENT | 26 | R/O | Set when PTP_SYNC interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_INTEGRITY_IDENT | 25 | R/O | Set when INTEGRITY interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_INJ_RDY1_IDENT | 21 | R/O | Set when INJ_RDY1 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_INJ_RDY0_IDENT | 20 | R/O | Set when INJ_RDY0 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_XTR_RDY1_IDENT | 17 | R/O | Set when XTR_RDY1 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_XTR_RDY0_IDENT | 16 | R/O | Set when XTR_RDY0 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |

*Table 605 •* **Fields in ICPU_IRQ1_IDENT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ICPU_IRQ1_BLK_ANA_IDENT | 15 | R/O | Set when BLK_ANA interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_DEV_ALL_IDENT | 14 | R/O | Set when DEV_ALL interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_SGPIO_IDENT | 13 | R/O | Set when SGPIO interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_GPIO_IDENT | 12 | R/O | Set when GPIO interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_TWI_IDENT | 11 | R/O | Set when TWI interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_FDMA_IDENT | 10 | R/O | Set when FDMA interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_TIMER2_IDENT | 9 | R/O | Set when TIMER2 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_TIMER1_IDENT | 8 | R/O | Set when TIMER1 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_TIMER0_IDENT | 7 | R/O | Set when TIMER0 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_UART_IDENT | 6 | R/O | Set when UART interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_PI_SD1_IDENT | 5 | R/O | Set when PI_SD1 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_PI_SD0_IDENT | 4 | R/O | Set when PI_SD0 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_SW1_IDENT | 3 | R/O | Set when SW1 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_SW0_IDENT | 2 | R/O | Set when SW0 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_EXT_IRQ1_IDENT | 1 | R/O | Set when EXT_IRQ1 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |
| ICPU_IRQ1_EXT_IRQ0_IDENT | 0 | R/O | Set when EXT_IRQ0 interrupt is a source of the ICPU_IRQ1 interrupt. | 0x0 |

### 7.15.4.10 ICPU_CFG:INTR:EXT_IRQ0_ENA

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 606 •* **Fields in EXT_IRQ0_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_ENA | 0 | R/W | Enables EXT_IRQ0 interrupt<br>0: Interrupt is disabled<br>1: Interrupt is enabled | 0x0 |

### 7.15.4.11 ICPU_CFG:INTR:EXT_IRQ0_IDENT

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Identifies the source(s) of an active interrupt on output interrupt: EXT_IRQ0. All asserted interrupts are shown as active high.

*Table 607* • **Fields in EXT_IRQ0_IDENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_MIIM1_IDENT | 28 | R/O | Set when MIIM1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_MIIM0_IDENT | 27 | R/O | Set when MIIM0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_PTP_SYNC_IDENT | 26 | R/O | Set when PTP_SYNC interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_INTEGRITY_IDENT | 25 | R/O | Set when INTEGRITY interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_INJ_RDY1_IDENT | 21 | R/O | Set when INJ_RDY1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_INJ_RDY0_IDENT | 20 | R/O | Set when INJ_RDY0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_XTR_RDY1_IDENT | 17 | R/O | Set when XTR_RDY1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_XTR_RDY0_IDENT | 16 | R/O | Set when XTR_RDY0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_BLK_ANA_IDENT | 15 | R/O | Set when BLK_ANA interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_DEV_ALL_IDENT | 14 | R/O | Set when DEV_ALL interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_SGPIO_IDENT | 13 | R/O | Set when SGPIO interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_GPIO_IDENT | 12 | R/O | Set when GPIO interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_TWI_IDENT | 11 | R/O | Set when TWI interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_FDMA_IDENT | 10 | R/O | Set when FDMA interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_TIMER2_IDENT | 9 | R/O | Set when TIMER2 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_TIMER1_IDENT | 8 | R/O | Set when TIMER1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_TIMER0_IDENT | 7 | R/O | Set when TIMER0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_UART_IDENT | 6 | R/O | Set when UART interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_PI_SD1_IDENT | 5 | R/O | Set when PI_SD1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |

*Table 607 •* **Fields in EXT_IRQ0_IDENT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_PI_SD0_IDENT | 4 | R/O | Set when PI_SD0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_SW1_IDENT | 3 | R/O | Set when SW1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_SW0_IDENT | 2 | R/O | Set when SW0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_EXT_IRQ1_IDENT | 1 | R/O | Set when EXT_IRQ1 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |
| EXT_IRQ0_EXT_IRQ0_IDENT | 0 | R/O | Set when EXT_IRQ0 interrupt is a source of the EXT_IRQ0 interrupt. | 0x0 |

### 7.15.4.12 ICPU_CFG:INTR:EXT_IRQ1_ENA

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 608 •* **Fields in EXT_IRQ1_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ1_ENA | 0 | R/W | Enables EXT_IRQ1 interrupt<br>0: Interrupt is disabled<br>1: Interrupt is enabled | 0x0 |

### 7.15.4.13 ICPU_CFG:INTR:EXT_IRQ1_IDENT

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Identifies the source(s) of an active interrupt on output interrupt: EXT_IRQ1. All asserted interrupts are shown as active high.

*Table 609 •* **Fields in EXT_IRQ1_IDENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ1_MIIM1_IDENT | 28 | R/O | Set when MIIM1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_MIIM0_IDENT | 27 | R/O | Set when MIIM0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_PTP_SYNC_IDENT | 26 | R/O | Set when PTP_SYNC interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_INTEGRITY_IDENT | 25 | R/O | Set when INTEGRITY interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_INJ_RDY1_IDENT | 21 | R/O | Set when INJ_RDY1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_INJ_RDY0_IDENT | 20 | R/O | Set when INJ_RDY0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |

*Table 609 •* **Fields in EXT_IRQ1_IDENT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ1_XTR_RDY1_IDENT | 17 | R/O | Set when XTR_RDY1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_XTR_RDY0_IDENT | 16 | R/O | Set when XTR_RDY0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_BLK_ANA_IDENT | 15 | R/O | Set when BLK_ANA interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_DEV_ALL_IDENT | 14 | R/O | Set when DEV_ALL interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_SGPIO_IDENT | 13 | R/O | Set when SGPIO interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_GPIO_IDENT | 12 | R/O | Set when GPIO interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_TWI_IDENT | 11 | R/O | Set when TWI interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_FDMA_IDENT | 10 | R/O | Set when FDMA interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_TIMER2_IDENT | 9 | R/O | Set when TIMER2 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_TIMER1_IDENT | 8 | R/O | Set when TIMER1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_TIMER0_IDENT | 7 | R/O | Set when TIMER0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_UART_IDENT | 6 | R/O | Set when UART interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_PI_SD1_IDENT | 5 | R/O | Set when PI_SD1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_PI_SD0_IDENT | 4 | R/O | Set when PI_SD0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_SW1_IDENT | 3 | R/O | Set when SW1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_SW0_IDENT | 2 | R/O | Set when SW0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_EXT_IRQ1_IDENT | 1 | R/O | Set when EXT_IRQ1 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |
| EXT_IRQ1_EXT_IRQ0_IDENT | 0 | R/O | Set when EXT_IRQ0 interrupt is a source of the EXT_IRQ1 interrupt. | 0x0 |

### 7.15.4.14 ICPU_CFG:INTR:DEV_IDENT

**Parent:** ICPU_CFG:INTR

**Instances:** 1

Shows the sources of the DEV_ALL interrupt.

*Table 610 •* **Fields in DEV_IDENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DEV_IDENT | 31:0 | R/O | Bits in this field is set when the corresponding device is interrupting, bit 0 corresponds to device 0, bit 1 to device 1 and so on. When any bit in this field is set the DEV_ALL interrupt is also asserted. | 0x00000000 |

### 7.15.4.15 ICPU_CFG:INTR:EXT_IRQ0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 611 •* **Fields in EXT_IRQ0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_INTR_DRV | 6 | R/W | Configures when to drive the external interrupt EXT_IRQ0 output, this setting applies only when EXT_IRQ0 is configured for output mode.<br>0: Only drive when interrupt is active<br>1: Always driven | 0x0 |
| EXT_IRQ0_INTR_DIR | 5 | R/W | Controls the direction of external interrupt: EXT_IRQ0.<br>In input mode the interrupt can used as source in the interrupt controller, in this mode any configurations related to the output mode of the interrupt has no effect.<br>In output mode sources can be assigned to the interrupt, in this mode the EXT_IRQ0 must not be enabled as interrupt source (INTR_ENA.EXT_IRQ0_INTR_ENA must remain 0).<br>0: Input<br>1: Output | 0x0 |
| EXT_IRQ0_INTR_POL | 4 | R/W | Controls the interrupt polarity of external interrupt: EXT_IRQ0. This setting is applies to both to input and output mode. The polarity is corrected at the edge of the chip, internally interrupts are always active-high.<br>0: Active low<br>1: Active high | 0x0 |

*Table 611 •* **Fields in EXT_IRQ0_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ0_INTR_FORCE | 3 | One-shot | Set to force assertion of EXT_IRQ0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| EXT_IRQ0_INTR_TRIGGER | 2 | R/W | Controls whether interrupts from the EXT_IRQ0 interrupt are edge (low-to-high-transition) or level (interrupt while high value is seen) sensitive.<br>0: LEVEL sensitive<br>1: EDGE sensitive | 0x0 |
| EXT_IRQ0_INTR_SEL | 1:0 | R/W | Selects the destination of the EXT_IRQ0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.16 ICPU_CFG:INTR:EXT_IRQ1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 612 •* **Fields in EXT_IRQ1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ1_INTR_DRV | 6 | R/W | Configures when to drive the external interrupt EXT_IRQ1 output, this setting applies only when EXT_IRQ1 is configured for output mode.<br>0: Only drive when interrupt is active<br>1: Always driven | 0x0 |
| EXT_IRQ1_INTR_DIR | 5 | R/W | Controls the direction of external interrupt: EXT_IRQ1.<br>In input mode the interrupt can used as source in the interrupt controller, in this mode any configurations related to the output mode of the interrupt has no effect.<br>In output mode sources can be assigned to the interrupt, in this mode the EXT_IRQ1 must not be enabled as interrupt source (INTR_ENA.EXT_IRQ1_INTR_ENA must remain 0).<br>0: Input<br>1: Output | 0x0 |

*Table 612 •* **Fields in EXT_IRQ1_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EXT_IRQ1_INTR_POL | 4 | R/W | Controls the interrupt polarity of external interrupt: EXT_IRQ1. This setting is applies to both to input and output mode. The polarity is corrected at the edge of the chip, internally interrupts are always active-high.<br>0: Active low<br>1: Active high | 0x0 |
| EXT_IRQ1_INTR_FORCE | 3 | One-shot | Set to force assertion of EXT_IRQ1 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| EXT_IRQ1_INTR_TRIGGER | 2 | R/W | Controls whether interrupts from the EXT_IRQ1 interrupt are edge (low-to-high-transition) or level (interrupt while high value is seen) sensitive.<br>0: LEVEL sensitive<br>1: EDGE sensitive | 0x0 |
| EXT_IRQ1_INTR_SEL | 1:0 | R/W | Selects the destination of the EXT_IRQ1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.17 ICPU_CFG:INTR:SW0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 613 •* **Fields in SW0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW0_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| SW0_INTR_FORCE | 3 | One-shot | Set to force assertion of SW0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 613 •* **Fields in SW0_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW0_INTR_SEL | 1:0 | R/W | Selects the destination of the SW0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.18 ICPU_CFG:INTR:SW1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 614 •* **Fields in SW1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SW1_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| SW1_INTR_FORCE | 3 | One-shot | Set to force assertion of SW1 interrupt. | 0x0 |
| SW1_INTR_SEL | 1:0 | R/W | Selects the destination of the SW1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.19 ICPU_CFG:INTR:MIIM1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 615 •* **Fields in MIIM1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM1_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| MIIM1_INTR_FORCE | 3 | One-shot | Set to force assertion of MIIM1 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| MIIM1_INTR_SEL | 1:0 | R/W | Selects the destination of the MIIM1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.20 ICPU_CFG:INTR:MIIM0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 616 •* **Fields in MIIM0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM0_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| MIIM0_INTR_FORCE | 3 | One-shot | Set to force assertion of MIIM0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 616 •* **Fields in MIIM0_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MIIM0_INTR_SEL | 1:0 | R/W | Selects the destination of the MIIM0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.21 ICPU_CFG:INTR:PI_SD0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 617 •* **Fields in PI_SD0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PI_SD0_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| PI_SD0_INTR_FORCE | 3 | One-shot | Set to force assertion of PI_SD0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| PI_SD0_INTR_SEL | 1:0 | R/W | Selects the destination of the PI_SD0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.22 ICPU_CFG:INTR:PI_SD1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 618 •* **Fields in PI_SD1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PI_SD1_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| PI_SD1_INTR_FORCE | 3 | One-shot | Set to force assertion of PI_SD1 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| PI_SD1_INTR_SEL | 1:0 | R/W | Selects the destination of the PI_SD1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.23 ICPU_CFG:INTR:UART_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 619 •* **Fields in UART_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UART_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| UART_INTR_FORCE | 3 | One-shot | Set to force assertion of UART interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 619 •* **Fields in UART_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| UART_INTR_SEL | 1:0 | R/W | Selects the destination of the UART interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.24 ICPU_CFG:INTR:TIMER0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 620 •* **Fields in TIMER0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER0_INTR_FORCE | 3 | One-shot | Set to force assertion of TIMER0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| TIMER0_INTR_SEL | 1:0 | R/W | Selects the destination of the TIMER0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.25 ICPU_CFG:INTR:TIMER1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 621 •* **Fields in TIMER1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER1_INTR_FORCE | 3 | One-shot | Set to force assertion of TIMER1 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| TIMER1_INTR_SEL | 1:0 | R/W | Selects the destination of the TIMER1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.26 ICPU_CFG:INTR:TIMER2_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 622 •* **Fields in TIMER2_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER2_INTR_FORCE | 3 | One-shot | Set to force assertion of TIMER2 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| TIMER2_INTR_SEL | 1:0 | R/W | Selects the destination of the TIMER2 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.27 ICPU_CFG:INTR:FDMA_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 623 •* **Fields in FDMA_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FDMA_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| FDMA_INTR_FORCE | 3 | One-shot | Set to force assertion of FDMA interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| FDMA_INTR_SEL | 1:0 | R/W | Selects the destination of the FDMA interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.28 ICPU_CFG:INTR:TWI_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 624 •* **Fields in TWI_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TWI_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| TWI_INTR_FORCE | 3 | One-shot | Set to force assertion of TWI interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| TWI_INTR_SEL | 1:0 | R/W | Selects the destination of the TWI interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.29 ICPU_CFG:INTR:GPIO_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 625 •* **Fields in GPIO_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GPIO_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| GPIO_INTR_FORCE | 3 | One-shot | Set to force assertion of GPIO interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 625 •* **Fields in GPIO_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GPIO_INTR_SEL | 1:0 | R/W | Selects the destination of the GPIO interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.30 ICPU_CFG:INTR:SGPIO_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 626 •* **Fields in SGPIO_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SGPIO_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| SGPIO_INTR_FORCE | 3 | One-shot | Set to force assertion of SGPIO interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| SGPIO_INTR_SEL | 1:0 | R/W | Selects the destination of the SGPIO interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.31 ICPU_CFG:INTR:DEV_ALL_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 627 •* **Fields in DEV_ALL_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DEV_ALL_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| DEV_ALL_INTR_FORCE | 3 | One-shot | Set to force assertion of DEV_ALL interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| DEV_ALL_INTR_SEL | 1:0 | R/W | Selects the destination of the DEV_ALL interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.32 ICPU_CFG:INTR:BLK_ANA_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 628 •* **Fields in BLK_ANA_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BLK_ANA_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| BLK_ANA_INTR_FORCE | 3 | One-shot | Set to force assertion of BLK_ANA interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 628 •* **Fields in BLK_ANA_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BLK_ANA_INTR_SEL | 1:0 | R/W | Selects the destination of the BLK_ANA interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.33 ICPU_CFG:INTR:XTR_RDY0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 629 •* **Fields in XTR_RDY0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| XTR_RDY0_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| XTR_RDY0_INTR_FORCE | 3 | One-shot | Set to force assertion of XTR_RDY0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| XTR_RDY0_INTR_SEL | 1:0 | R/W | Selects the destination of the XTR_RDY0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.34 ICPU_CFG:INTR:XTR_RDY1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 630 •* **Fields in XTR_RDY1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| XTR_RDY1_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| XTR_RDY1_INTR_FORCE | 3 | One-shot | Set to force assertion of XTR_RDY1 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| XTR_RDY1_INTR_SEL | 1:0 | R/W | Selects the destination of the XTR_RDY1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.35 ICPU_CFG:INTR:INJ_RDY0_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 631 •* **Fields in INJ_RDY0_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_RDY0_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| INJ_RDY0_INTR_FORCE | 3 | One-shot | Set to force assertion of INJ_RDY0 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 631 •* **Fields in INJ_RDY0_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_RDY0_INTR_SEL | 1:0 | R/W | Selects the destination of the INJ_RDY0 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.36 ICPU_CFG:INTR:INJ_RDY1_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 632 •* **Fields in INJ_RDY1_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_RDY1_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| INJ_RDY1_INTR_FORCE | 3 | One-shot | Set to force assertion of INJ_RDY1 interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| INJ_RDY1_INTR_SEL | 1:0 | R/W | Selects the destination of the INJ_RDY1 interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.37 ICPU_CFG:INTR:INTEGRITY_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 633 •* **Fields in INTEGRITY_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INTEGRITY_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| INTEGRITY_INTR_FORCE | 3 | One-shot | Set to force assertion of INTEGRITY interrupt. This field is cleared immediately after generating interrupt. | 0x0 |
| INTEGRITY_INTR_SEL | 1:0 | R/W | Selects the destination of the INTEGRITY interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

### 7.15.4.38 ICPU_CFG:INTR:PTP_SYNC_INTR_CFG

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 634 •* **Fields in PTP_SYNC_INTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_SYNC_INTR_BYPASS | 4 | R/W | Set to bypass sticky interrupt functionality. When set the value from the interrupting source is passed directly to the destination interrupt. This feature can be useful when mapping a small number of interrupts via external interrupt output to an external CPU.<br>When this field is set, the TRIGGER and FORCE fields no longer has any effect. | 0x0 |
| PTP_SYNC_INTR_FORCE | 3 | One-shot | Set to force assertion of PTP_SYNC interrupt. This field is cleared immediately after generating interrupt. | 0x0 |

*Table 634 •* **Fields in PTP_SYNC_INTR_CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTP_SYNC_INTR_SEL | 1:0 | R/W | Selects the destination of the PTP_SYNC interrupt.<br>0: ICPU_IRQ0<br>1: ICPU_IRQ1<br>2: EXT_IRQ0<br>3: EXT_IRQ1 | 0x0 |

#### 7.15.4.39 ICPU_CFG:INTR:DEV_ENA

**Parent:** ICPU_CFG:INTR

**Instances:** 1

*Table 635 •* **Fields in DEV_ENA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DEV_ENA | 31:0 | R/W | Clear individual bits in this register to disable interrupts from specific devices. | 0x00000000 |

### 7.15.5 ICPU_CFG:GPDMA

**Parent:** ICPU_CFG

**Instances:** 1

*Table 636 •* **Registers in GPDMA**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| FDMA_CFG | 0x00000000 | 1 | Common Injection or Extraction Configuration | Page 484 |
| FDMA_CH_CFG | 0x00000008 | 8<br>0x00000004 | FDMA Channel Usage and Flow Control | Page 485 |
| FDMA_INJ_CFG | 0x00000028 | 2<br>0x00000004 | FDMA Injection Parameters | Page 485 |
| FDMA_XTR_CFG | 0x00000030 | 2<br>0x00000004 | FDMA Extraction Parameters | Page 486 |
| FDMA_XTR_STAT_LAST_DCB | 0x00000038 | 2<br>0x00000004 | Extraction Status for FDMA Engine | Page 487 |
| FDMA_FRM_CNT | 0x00000040 | 1 | Frame Counter and Flow Control Status | Page 487 |
| FDMA_BP_TO_INT | 0x00000044 | 1 | FDMA Backpressure Timeout Interrupt | Page 488 |
| FDMA_BP_TO_DIV | 0x00000048 | 1 | FDMA Timeout Divider | Page 488 |

#### 7.15.5.1 ICPU_CFG:GPDMA:FDMA_CFG

**Parent:** ICPU_CFG:GPDMA

**Instances:** 1

*Table 637 •* **Fields in FDMA_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_GRP_ABRT_ID | 2 | R/W | Specifies an injection group ID to abort frames on when setting INJ_GRP_ABRT. This field may only be changed when INJ_GRP_ABRT is cleared. | 0x0 |
| INJ_GRP_ABRT | 1 | One-shot | Set to abort the frame currently being transmitted on the injection group indicated by INJ_GRP_ABRT_ID. This field is cleared once the abort has been accepted. If no frame is currently being transmitted (on the injection group) then no aborting will occur. | 0x0 |
| FDMA_ENA | 0 | R/W | Enable FDMA access to the queuing system.When this field is set, manual injection and extraction must not be done through the DEVCPU registers. | 0x0 |

### 7.15.5.2 ICPU_CFG:GPDMA:FDMA_CH_CFG

**Parent:** ICPU_CFG:GPDMA

**Instances:** 8

Configurations for each of the DMA channels.

*Table 638 •* **Fields in FDMA_CH_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| USAGE | 1 | R/W | Controls the usage of the channel. The channel can be configured for either frame extraction (XTR) or frame injection (INJ)<br>0: The channel is an extraction channel (XTR)<br>1: The channel is an injection channel (INJ) | 0x0 |
| CH_ENA | 0 | R/W | Enable channel for the specified function. | 0x0 |

### 7.15.5.3 ICPU_CFG:GPDMA:FDMA_INJ_CFG

**Parent:** ICPU_CFG:GPDMA

**Instances:** 2

Configurations for each of the injection groups.

*Table 639 •* **Fields in FDMA_INJ_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_GRP_BP_TO_INT_ENA | 4 | R/W | Set this field to enable back pressure timeout interrupt for this injection group, see FDMA_BP_TIMEOUT_INT:INJ_BP_TIMEOUT_INT for more information. | 0x0 |
| INJ_GRP_BP_ENA | 3 | R/W | Enable back pressure from the corresponding injection channel. If an injection channel is used this field (and INJ_GRP_BP_MAP) must be set.<br>0: Back-pressure is disabled.<br>1: Back-pressure is enabled. | 0x0 |
| INJ_GRP_BP_MAP | 2:0 | R/W | To correctly generate backpressure to the DMA from individual injection groups, configure the DMA channel ID which may send frames to the corresponding injection group. If the injection group is not used then this field is a don't-care.<br>Note that an injection group can only receive frames from a single DMA channel, while DMA channels can inject to multiple injection groups. When a DMA channel injects to multiple injection groups, backpressure must be enabled from all of the injection groups. | 0x0 |

### 7.15.5.4 ICPU_CFG:GPDMA:FDMA_XTR_CFG

**Parent:** ICPU_CFG:GPDMA

**Instances:** 2

Configurations for each of the extraction groups.

*Table 640 •* **Fields in FDMA_XTR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| XTR_BURST_SIZE | 2:0 | R/W | Must be configured to the same value as CTL0:SRC_MSIZE for the corresponding DMA channel.<br>0 : 1<br>1 : 4<br>2 : 8<br>3 : 16<br>4 : 32<br>5 : 64<br>6-7 : reserved, do not use | 0x1 |

### 7.15.5.5 ICPU_CFG:GPDMA:FDMA_XTR_STAT_LAST_DCB

**Parent:** ICPU_CFG:GPDMA

**Instances:** 2

ThIs register provides the extraction status to be used by this FDMA engine.

*Table 641 •* **Fields in FDMA_XTR_STAT_LAST_DCB**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| XTR_STAT_FRM_LEN | 31:16 | R/O | Length of frame (in bytes).<br>If frames are spread across multiple DCBs this field is incremental; it shows the number of bytes written to the current and all previous DCBs, at the last DCB (EOF when is set), then value then represents the total frame-length. | 0x0000 |
| XTR_STAT_ABORT | 4 | R/O | Frame has been aborted, this will happen if frame is longer than maximum allowed size. | 0x0 |
| XTR_STAT_PRUNED | 3 | R/O | Frame has been pruned (see extraction queue registers for more details).<br>0: Not pruned<br>1: Pruned | 0x0 |
| XTR_STAT_EOF | 2 | R/O | End of frame<br>0: Not EOF<br>1: EOF | 0x0 |
| XTR_STAT_SOF | 1 | R/O | Start of frame<br>0: Not SOF<br>1: SOF | 0x0 |
| XTR_STAT_VLD | 0 | R/O | Always reads as '1'. | 0x1 |

### 7.15.5.6 ICPU_CFG:GPDMA:FDMA_FRM_CNT

**Parent:** ICPU_CFG:GPDMA

**Instances:** 1

*Table 642 •* **Fields in FDMA_FRM_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FDMA_FRM_CNT | 15:0 | R/W | This counter is incremented by 1 for every frame that is moved through the FDMA (both XTR or INJ). The counter increments when end-of-frame is processed by the FDMA. | 0x0000 |

### 7.15.5.7 ICPU_CFG:GPDMA:FDMA_BP_TO_INT

**Parent:** ICPU_CFG:GPDMA

**Instances:** 1

As long as a field in this register is set, the FDMA will indicate interrupt towards the interrupt controller.

*Table 643 •* **Fields in FDMA_BP_TO_INT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_BP_TO_INT | 1:0 | Sticky | This is an indication of backpressure timeout interrupt. If a bit in this field is set the corresponding injection group has been in back-pressure for more than the allowed time (as configured in FDMA_BP_TO_DIV:INJ_BP_TO_DIV). Enable backpressure timeout interrupt in FDMA_INJ_CFG:INJ_GRP_BP_TO_INT_ENA. | 0x0 |

### 7.15.5.8 ICPU_CFG:GPDMA:FDMA_BP_TO_DIV

**Parent:** ICPU_CFG:GPDMA

**Instances:** 1

*Table 644 •* **Fields in FDMA_BP_TO_DIV**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INJ_BP_TO_DIV_RLOAD | 16 | One-shot | Set this field to force reload of the backpressure timeout divider. | 0x0 |
| INJ_BP_TO_DIV | 15:0 | R/W | Configures the timeout for injection group backpressure interrupt. The timeout is calculated as follows: timeout(s) = <div-value>/(sysfrequency(MHz) * 244) E.g. configuring a timeout value of 1220 in a 200MHz system yields a timeout of 25ms. | 0x04C4 |

## 7.15.6 ICPU_CFG:INJ_FRM_SPC

**Parent:** ICPU_CFG

**Instances:** 8

*Table 645 •* **Registers in INJ_FRM_SPC**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| INJ_FRM_SPC_TMR | 0x00000000 | 1 | Injection frame spacing timer | Page 489 |
| INJ_FRM_SPC_TMR_CFG | 0x00000004 | 1 | Reload value for injection frame spacing timer | Page 489 |
| INJ_FRM_SPC_LACK_CNTR | 0x00000008 | 1 | Lack counter | Page 490 |
| INJ_FRM_SPC_CFG | 0x0000000C | 1 | Injection frame spacing configuration register | Page 490 |

### 7.15.6.1 ICPU_CFG:INJ_FRM_SPC:INJ_FRM_SPC_TMR

**Parent:** ICPU_CFG:INJ_FRM_SPC

**Instances:** 1

*Table 646 •* **Fields in INJ_FRM_SPC_TMR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TMR | 31:0 | R/O | The "frame space" timer, enabled when INJ_FRM_SPC_CONFIG.TMR_ENA is set. When it reaches zero, it provides a tick to INJ_FRM_LACK_CNTR, and reloads the value held in INJ_FRM_SPC_TMR_CFG. The counter is down-counting. The resulting delay between frames is (n+1)*ahb_clk_p where n is the timer reload value and ahb_clk_p is the clock period of the ahb bus. | 0x00000000 |

### 7.15.6.2 ICPU_CFG:INJ_FRM_SPC:INJ_FRM_SPC_TMR_CFG

**Parent:** ICPU_CFG:INJ_FRM_SPC

**Instances:** 1

*Table 647 •* **Fields in INJ_FRM_SPC_TMR_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TMR_CFG | 31:0 | R/W | Reload value for INJ_FRM_SPC_TMR. | 0x00000000 |

### 7.15.6.3 ICPU_CFG:INJ_FRM_SPC:INJ_FRM_SPC_LACK_CNTR

**Parent:** ICPU_CFG:INJ_FRM_SPC

**Instances:** 1

***Table 648 •*** **Fields in INJ_FRM_SPC_LACK_CNTR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LACK_CNTR | 7:0 | R/W | When INJ_FRM_SPC_CFG.FRM_SPC_ENA is set, this counter counts the number of ticks provided by the INJ_FRM_SPC_TMR and is decremented by hardware for every transmitted frame. In other words, the value of lack counter value is the number of frames which it is OK to transmit unspaced. Is used in conjunction with the queue-system fill-level to signal to the DMA that it is OK to transmit the next frame. | 0x00 |

### 7.15.6.4 ICPU_CFG:INJ_FRM_SPC:INJ_FRM_SPC_CFG

**Parent:** ICPU_CFG:INJ_FRM_SPC

**Instances:** 1

***Table 649 •*** **Fields in INJ_FRM_SPC_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FRM_SPC_ENA | 0 | R/W | This bit is used to generally enable/disable the frame spacing feature. | 0x0 |
| TMR_ENA | 1 | R/W | Controls whether the INJ_FRM_SPC_TMR is counting or not. When this field is 0 the reload value is written to the frame space timer and the timer is not running. When this field is 1 the timer is running and is reloaded when it reaches zero. | 0x0 |

## 7.15.7 ICPU_CFG:TIMERS

**Parent:** ICPU_CFG

**Instances:** 1

*Table 650 •* **Registers in TIMERS**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| WDT | 0x00000000 | 1 | Watchdog Timer | Page 491 |
| TIMER_TICK_DIV | 0x00000004 | 1 | Timer Tick Divider | Page 492 |
| TIMER_VALUE | 0x00000008 | 3<br>0x00000004 | Timer value | Page 492 |
| TIMER_RELOAD_VALUE | 0x00000014 | 3<br>0x00000004 | Timer Reload Value | Page 493 |
| TIMER_CTRL | 0x00000020 | 3<br>0x00000004 | Timer Control | Page 493 |

### 7.15.7.1 ICPU_CFG:TIMERS:WDT

**Parent:** ICPU_CFG:TIMERS

**Instances:** 1

*Table 651 •* **Fields in WDT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WDT_STATUS | 9 | R/O | Shows whether the last reset was caused by a watchdog timer reset. This field is updated during reset, therefore it is always valid.<br>0: Reset was not caused by WDT<br>1: Reset was caused by WDT timeout | 0x0 |
| WDT_ENABLE | 8 | R/W | Use this field to enable or disable the watchdog timer. When the WDT is enabled, it causes a reset after 2 seconds if it is not periodically reset. This field is only read by the WDT after a successful lock sequence (WDT_LOCK).<br>0: WDT is disabled<br>1: WDT is enabled | 0x0 |
| WDT_LOCK | 7:0 | R/W | Use this field to configure and reset the WDT. When writing 0xBE to this field immediately followed by writing 0xEF, the WDT resets and configurations are read from this register (as set when the 0xEF is written). When the WDT is enabled, writing any value other than 0xBE or 0xEF after 0xBE is written, causes a WDT reset as if the timer had run out. | 0x00 |

### 7.15.7.2 ICPU_CFG:TIMERS:TIMER_TICK_DIV

**Parent:** ICPU_CFG:TIMERS

**Instances:** 1

*Table 652* • **Fields in TIMER_TICK_DIV**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER_TICK_DIV | 17:0 | R/W | The timer tick generator runs from a 250MHz base clock. By default, the divider value generates a timer tick every 100 us (10 KHz). The timer tick is used for all of the timers (except the WDT). This field must not be set to generate a timer tick of less than 0.1 us (higher than 10 MHz). If this field is changed, it may take up to 2 ms before the timers are running stable at the new frequency.<br>The timer tick frequency is: 250MHz/(TIMER_TICK_DIV+1). | 0x061A7 |

### 7.15.7.3 ICPU_CFG:TIMERS:TIMER_VALUE

**Parent:** ICPU_CFG:TIMERS

**Instances:** 3

*Table 653 •* **Fields in TIMER_VALUE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER_VAL | 31:0 | R/W | The current value of the timer. When enabled via TIMER_CTRL.TIMER_ENA the timer decrements at every timer tick (see TIMER_TICK_DIV for more info on timer tick frequency). When the timer has reached 0, and a timer-tick is received, then an interrupt is generated. For example; If a periodic interrupt is needed every 1ms, and the timer tick is generated every 100us then the TIMER_VALUE (and TIMER_RELOAD_VALUE) must be configured to 9. By default the timer will reload from the TIMER_RELOAD_VALUE when interrupt is generated, and then continue decrementing from the reloaded value. It is possible to make the timer stop after generating interrupt by setting TIMER_CTRL.ONE_SHOT. | 0x00000000 |

### 7.15.7.4 ICPU_CFG:TIMERS:TIMER_RELOAD_VALUE

**Parent:** ICPU_CFG:TIMERS

**Instances:** 3

*Table 654 •* **Fields in TIMER_RELOAD_VALUE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RELOAD_VAL | 31:0 | R/W | The contents of this field are loaded into the corresponding timer (TIMER_VALUE) when it wraps (decrements a zero). | 0x00000000 |

### 7.15.7.5 ICPU_CFG:TIMERS:TIMER_CTRL

**Parent:** ICPU_CFG:TIMERS

**Instances:** 3

*Table 655 •* **Fields in TIMER_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ONE_SHOT_ENA | 2 | R/W | When set the timer will automatically disable itself after it has generated interrupt. | 0x0 |

*Table 655 •* **Fields in TIMER_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TIMER_ENA | 1 | R/W | When enabled, the corresponding timer decrements at each timer-tick. If TIMER_CTRL.ONE_SHOT_ENA is set this field is cleared when the timer reach 0 and interrupt is generated.<br>0: Timer is disabled<br>1: Timer is enabled | 0x0 |
| FORCE_RELOAD | 0 | One-shot | Set this field to force the reload of the timer, this will set the TIMER_VALUE to TIMER_RELOAD_VALUE for the corresponding timer. This field can be set at the same time as enabling the counter, in that case the counter will be reloaded and then enabled for counting. | 0x0 |

## 7.15.8 ICPU_CFG:MEMCTRL

**Parent:** ICPU_CFG

**Instances:** 1

*Table 656 •* **Registers in MEMCTRL**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MEMCTRL_CTRL | 0x00000000 | 1 | Control register | Page 495 |
| MEMCTRL_CFG | 0x00000004 | 1 | Configuration register | Page 495 |
| MEMCTRL_STAT | 0x00000008 | 1 | Status register | Page 496 |
| MEMCTRL_REF_PERIOD | 0x0000000C | 1 | Refresh period configuration | Page 496 |
| MEMCTRL_TIMING0 | 0x00000014 | 1 | Timing register 0 | Page 497 |
| MEMCTRL_TIMING1 | 0x00000018 | 1 | Timing register 1 | Page 498 |
| MEMCTRL_TIMING2 | 0x0000001C | 1 | Timing register 2 | Page 499 |
| MEMCTRL_TIMING3 | 0x00000020 | 1 | Timing register 3 | Page 499 |
| MEMCTRL_MR0_VAL | 0x00000024 | 1 | Mode Register 0 Value | Page 500 |
| MEMCTRL_MR1_VAL | 0x00000028 | 1 | Mode Register 1/Extended Mode Register Value | Page 500 |
| MEMCTRL_MR2_VAL | 0x0000002C | 1 | Mode Register 2/Extended Mode Register 2 Value | Page 501 |
| MEMCTRL_MR3_VAL | 0x00000030 | 1 | Mode Register 3/Extended Mode Register 3 Value | Page 501 |
| MEMCTRL_TERMRES_CTRL | 0x00000034 | 1 | TBA | Page 501 |

*Table 656 •* **Registers in MEMCTRL *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| MEMCTRL_DQS_DLY | 0x0000003C | 1 | DQS window configuration | Page 502 |
| MEMCTRL_DQS_AUTO | 0x00000040 | 1 | DQS window automatic drift detect/adjust | Page 503 |
| MEMPHY_CFG | 0x00000044 | 1 | Control register | Page 503 |
| MEMPHY_ZCAL | 0x00000060 | 1 | Impedance calibration | Page 504 |

### 7.15.8.1 ICPU_CFG:MEMCTRL:MEMCTRL_CTRL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 657 •* **Fields in MEMCTRL_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STALL_REF_ENA | 1 | R/W | Set this field to postpone refresh of the SDRAM for as long as possible. Refresh will not be initiated until the number of pending refreshes reaches MEMCTRL_REF_PERIOD.MAX_PEND_REF.<br>Interrupt routines and other high-priority tasks can set this field to ensure uninterrupted access to the memory. | 0x0 |
| INITIALIZE | 0 | One-shot | Set this field to force the memory controller to initialize the SDRAM. This field is automatically cleared after the initialization sequence is complete.<br>Note: All other memory controller registers must have been configured appropriately before setting this field. | 0x0 |

### 7.15.8.2 ICPU_CFG:MEMCTRL:MEMCTRL_CFG

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 658 •* **Fields in MEMCTRL_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BURST_SIZE | 10 | R/W | The number of data-bytes that is transmitted during one burst (of the defined burst length: BURST_LEN).<br>0: 8 data-bytes per burst.<br>1: 16 data-bytes per burst. | 0x0 |
| BURST_LEN | 9 | R/W | The burst size that is used by the SDRAM controller. The SDRAM must be configured with the corresponding burst size (through the MEMCTRL_MDSET_VAL register.)<br>Note: The number of data-bytes that is transmitted during one burst must be encoded in the BURST_SIZE field.<br>0 : BURST4<br>1 : BURST8 | 0x0 |
| BANK_CNT | 8 | R/W | Number of banks in the SDRAM configuration being used.<br>0 : 4 banks<br>1 : 8 banks | 0x0 |
| MSB_ROW_ADDR | 7:4 | R/W | This field should be programmed to 1 less than the number of row address bits for the SDRAM configuration in use. | 0x0 |
| MSB_COL_ADDR | 3:0 | R/W | This field should be programmed to 1 less than the number of column address bits for the SDRAM configuration in use. | 0x0 |

### 7.15.8.3 ICPU_CFG:MEMCTRL:MEMCTRL_STAT

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 659 •* **Fields in MEMCTRL_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INIT_DONE | 0 | R/O | This field is set after initialization of the SDRAM is done. | 0x0 |

### 7.15.8.4 ICPU_CFG:MEMCTRL:MEMCTRL_REF_PERIOD

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 660 •* **Fields in MEMCTRL_REF_PERIOD**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MAX_PEND_REF | 19:16 | R/W | Maximum number of refreshes that are allowed to be outstanding at any time. If the number of outstanding refreshes reaches this value, the memory controller will stop the data transfer in progress, issue the required number of refreshes and then continue. This field must not be set to 0 (will disable the controller). | 0x1 |
| REF_PERIOD | 15:0 | R/W | Refresh interval of the SDRAM expressed in terms of number of clock cycles. This value is calculated by dividing the average periodic refresh interval (tREFI) by the clock period. | 0x0100 |

### 7.15.8.5 ICPU_CFG:MEMCTRL:MEMCTRL_TIMING0

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

All asynchronous timing delays should be converted into the equivalent number of core-clocks (round up). Note: The SDRAM datasheet may specify parameters in a number of tCK cycles these are the same as clock cycles.

*Table 661 •* **Fields in MEMCTRL_TIMING0**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RD_TO_WR_DLY | 31:28 | R/W | Suggested value is 4. Value of 4 gives 2 cycles turn around time between the last read from the SDRAM and the first write to the SDRAM. | 0x4 |
| RESERVED | 27:24 | R/W | Must be set to its default. | 0x3 |
| RESERVED | 23:20 | R/W | Must be set to its default. | 0x2 |
| RAS_TO_PRECH_DLY | 19:16 | R/W | tRAS - 1 clock. Minimum delay between RAS and precharge commands. | 0x0 |
| WR_TO_PRECH_DLY | 15:12 | R/W | This value depends on the burst length used by the configuration.<br>BURST4: CL + tWR.<br>BURST8: CL + 2 + tWR.<br>Minimum delay between write and precharge commands. | 0x0 |

*Table 661 •* **Fields in MEMCTRL_TIMING0** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RD_TO_PRECH_DLY | 11:8 | R/W | This value depends on the burst length used by the configuration.<br>BURST4: 1.<br>BURST8: 3.<br>Minimum delay between read and precharge commands. | 0x0 |
| WR_DATA_XFR_DLY | 7:4 | R/W | CL - 3. Delay between the issue of a write command and when the data is transmitted. CL must not be less than 3 (this register cannot be configured to less than 0). | 0x0 |
| RD_DATA_XFR_DLY | 3:0 | R/W | This field should be programmed to 1. The receive window is also adjusted by the DQS drift detection logic, which adds an additional delay on top of this value. | 0x0 |

### 7.15.8.6 ICPU_CFG:MEMCTRL:MEMCTRL_TIMING1

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

All asynchronous timing delays should be converted into the equivalent number of core-clocks (round up). Note: The SDRAM datasheet may specify parameters in a number of tCK cycles these are the same as clock cycles.

*Table 662 •* **Fields in MEMCTRL_TIMING1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RAS_TO_RAS_SAME_BANK_DLY | 31:24 | R/W | tRC - 1. Minimum delay between successive open commands to the same bank. | 0x00 |
| BANK8_FAW_DLY | 23:16 | R/W | tFAW - 1 for an 8-bank DDR2 SDRAM. 0 for a 4-bank DDR2 SDRAM. For 8 bank DDR2 SDRAM configurations; this value specifies an additional row opening restriction when a fifth bank is opened consecutively after 4 banks have been opened with minimum tRRD on the same chip select. | 0x00 |
| PRECH_TO_RAS_DLY | 15:12 | R/W | tRP - 1. Minimum delay between issuing a precharge command and a RAS command to the same bank. | 0x0 |
| RAS_TO_RAS_DLY | 11:8 | R/W | tRRD - 1. Minimum delay between two RAS commands issued to the same chip select. | 0x0 |

*Table 662 •* **Fields in MEMCTRL_TIMING1** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RAS_TO_CAS_DLY | 7:4 | R/W | tRCD - AL - 1. Minimum delay between issuing of a RAS command and a CAS command to the same bank. | 0x0 |
| WR_TO_RD_DLY | 3:0 | R/W | BURST4: CL + tWTR, where tWTR converted to clock cycles must be at least 2.<br>BURST8: CL + 2 + tWTR, where tWTR converted to clock cycles must be at least 2.<br>Minimum delay from a write to a read command. | 0x0 |

### 7.15.8.7 ICPU_CFG:MEMCTRL:MEMCTRL_TIMING2

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

All asynchronous timing delays should be converted into the equivalent number of core-clocks (round up). Note: The SDRAM datasheet may specify parameters in a number of tCK cycles these are the same as clock cycles.

*Table 663 •* **Fields in MEMCTRL_TIMING2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PRECH_ALL_DLY | 31:28 | R/W | tRP - 1 for 4 bank memory and tRPA - 1 for 8 bank memory. Minimum delay between issuing a precharge all command and a LM/RAS command to any bank. | 0x0 |
| MDSET_DLY | 27:24 | R/W | tMRD - 1. Minimum delay required after a mode set command and before issuing any other command. | 0x0 |
| REF_DLY | 23:16 | R/W | tRFC - 1. Minimum delay between issuing of a refresh command and a RAS command. This value is assumed to be less than 67 clocks. | 0x00 |
| FOUR_HUNDRED_NS_DLY | 15:0 | R/W | Four hundred nanoseconds expressed in clock periods (round up). This is used during the initialization sequence. | 0x0000 |

### 7.15.8.8 ICPU_CFG:MEMCTRL:MEMCTRL_TIMING3

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

All asynchronous timing delays should be converted into the equivalent number of core-clocks (round up). Note: The SDRAM datasheet may specify parameters in a number of tCK cycles these are the same as clock cycles.

*Table 664 •* **Fields in MEMCTRL_TIMING3**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ODT_WR_DLY | 11:8 | R/W | Value to be used is AL + CL - 4. Number of clocks after the write command that the ODT signal for the SDRAM should be turned on. This implies that AL + CL should be greater than or equal to 4. | 0x0 |
| LOCAL_ODT_RD_DLY | 7:4 | R/W | Value to be used is MEMCTRL_TIMING0.RD_DATA_XFR_DLY. Number of clocks after the read command to enable of local on-die-termination (ODT). This delay is also adjusted by the DQS drift detection logic, which adds an additional delay on top of this value. | 0x0 |
| WR_TO_RD_CS_CHANGE_DLY | 3:0 | R/W | AL + CL - 1 but no less than 3. Minimum delay between a write command issued to one chip select followed by a read command to the other chip select. This value is less than the MEMCTRL_TIMING1:WR_TO_RD_DLY. | 0x0 |

### 7.15.8.9 ICPU_CFG:MEMCTRL:MEMCTRL_MR0_VAL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

Note: The memory controller will automatically generate the required bank-addresses used for addressing the different mode registers. Do not specify bank-addresses in this register.

*Table 665 •* **Fields in MEMCTRL_MR0_VAL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MR0_VAL | 15:0 | R/W | Value to be programmed into the mode register (0) during SDRAM initialization.<br>Bit 8 (DLL Reset) of this register must be set to 0, the memory controller automatically sets this bit when required during the initialization procedure. | 0x0000 |

### 7.15.8.10 ICPU_CFG:MEMCTRL:MEMCTRL_MR1_VAL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

Note: The memory controller will automatically generate the required bank-addresses used for addressing the different mode registers. Do not specify bank-addresses in this register.

*Table 666* • **Fields in MEMCTRL_MR1_VAL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MR1_VAL | 15:0 | R/W | Value to be programmed into mode register 1/extended mode register during SDRAM initialization.<br>Bits 7 through 9 (OCD Calibration Program) of this register must be set to 0x7. The memory controller set this field when required during the initialization procedure. | 0x0000 |

### 7.15.8.11 ICPU_CFG:MEMCTRL:MEMCTRL_MR2_VAL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

Note: The memory controller will automatically generate the required bank-addresses used for addressing the different mode registers. Do not specify bank-addresses in this register.

*Table 667* • **Fields in MEMCTRL_MR2_VAL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MR2_VAL | 15:0 | R/W | Value to be programmed into mode register 2/extended mode register 2 during SDRAM initialization. | 0x0000 |

### 7.15.8.12 ICPU_CFG:MEMCTRL:MEMCTRL_MR3_VAL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

Note: The memory controller will automatically generate the required bank-addresses used for addressing the different mode registers. Do not specify bank-addresses in this register.

*Table 668* • **Fields in MEMCTRL_MR3_VAL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MR3_VAL | 15:0 | R/W | Value to be programmed into mode register 3/extended mode register 3 during SDRAM initialization. | 0x0000 |

### 7.15.8.13 ICPU_CFG:MEMCTRL:MEMCTRL_TERMRES_CTRL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 669 •* **Fields in MEMCTRL_TERMRES_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ODT_WR_EXT | 3 | R/W | Set this field to extend the ODT termination output by one clock during write operations. | 0x0 |
| ODT_WR_ENA | 2 | R/W | Enables external termination during write operations. | 0x0 |
| LOCAL_ODT_RD_EXT | 1 | R/W | Set this field to extend the local termination by one clock during read operations. | 0x0 |
| LOCAL_ODT_RD_ENA | 0 | R/W | Enables local termination during a read operation. | 0x0 |

### 7.15.8.14 ICPU_CFG:MEMCTRL:MEMCTRL_DQS_DLY

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

This register is replicated two times, once for each Byte Lane (first replication corresponds to Byte Lane 0).

After initialization of the DRAM memory controller the read-data-path must be trained. This is needed so that the controller knows exactly when to sample read-data from the DRAM(s). During training a window of DQS_DLY settings is determined during which correct read-data is returned from the DRAM(s), after finding the window the mid-window-value (round down) is programmed into DQS_DLY and then auto-adjusting is enabled by setting MEMCTRL_DQS_AUTO:DQS_AUTO_ENA. Training is done per Byte-Lane, two DRAM addresses are needed for training (a low and a high address), the actual addresses depends on the number of byte-lanes in the system, and which byte-lane that is trained: In a system with _one_ byte lane (x8), addresses 0x0 and 0xF is used. In a system with _two_ byte lanes (x16), DRAM addresses 0x0 and 0xE is used for training Byte Lane 0, and addresses 0x1 and 0xF is used for training Byte Lane 1.

Training is done for in the following steps:

1) Clear DRAM addresses 0x0 through 0xF by writing 0x00 to each address.

2) Write 0xFF to both the low and the high DRAM address (the actual addresses are defined in the above section) .

3) Find the lower DQS_DLY limit by sweeping through delay settings (DQS_DLY, starting from 0x0) while reading the high DRAM address. Continue sweeping (incrementing DQS_DLY) until 0xFF is returned when reading the high address.

4) Find the upper DQS_DLY limit by continuing the sweep through delay settings (starting at the lower limit determined during step 3) while reading the low DRAM address. Continue sweeping (incrementing DQS_DLY) until reading from the low address no longer returns 0xFF. The upper limit is then the current DQS_DLY - 1.

*Table 670 •* **Fields in MEMCTRL_DQS_DLY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 10:8 | R/W | Must be set to its default. | 0x3 |
| RESERVED | 7:5 | R/W | Must be set to its default. | 0x3 |

*Table 670 •* **Fields in MEMCTRL_DQS_DLY *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DQS_DLY | 4:0 | R/W | This field configures read-window delay as an offset in 1/4 clock cycles from the fixed read-delay configured in MEMCTRL_TIMING0:RD_DATA_XFR_DLY. | 0x00 |

### 7.15.8.15 ICPU_CFG:MEMCTRL:MEMCTRL_DQS_AUTO

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

This register is subjected to the same replication scheme and encoding as MEMCTRL_DQS_DLY.

*Table 671 •* **Fields in MEMCTRL_DQS_AUTO**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DQS_AUTO_ENA | 0 | R/W | Set this field to enable automatic detection of drifting read-data-window. Drifting of the DQS read window occurs as the chip is heating/cooling. When this field is set MEMCTRL_DQS_DLY.DQS_DLY field will automatically be adjusted when a drift is detected by the hardware. | 0x0 |

### 7.15.8.16 ICPU_CFG:MEMCTRL:MEMPHY_CFG

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 672 •* **Fields in MEMPHY_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PHY_ODT_OE | 4 | R/W | Set to enable output drive of the ODT output. | 0x0 |
| PHY_CK_OE | 3 | R/W | Set to enable output drive of the CK/nCK and CKE outputs. | 0x0 |
| PHY_CL_OE | 2 | R/W | Set to enable output drive of the Command Lane outputs. | 0x0 |
| PHY_SSTL_ENA | 1 | R/W | Set this field to enable the SSTL drivers/receivers in the memory controllers physical interface. | 0x0 |
| PHY_RST | 0 | R/W | Master reset to the memory controller physical interface.<br>0: PHY is in working mode.<br>1: PHY is forced in reset. | 0x1 |

### 7.15.8.17 ICPU_CFG:MEMCTRL:MEMPHY_ZCAL

**Parent:** ICPU_CFG:MEMCTRL

**Instances:** 1

*Table 673 •* **Fields in MEMPHY_ZCAL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ZCAL_PROG_ODT | 8:5 | R/W | Together with the external reference resistor this field configures the SSTL On-Die-Termination (ODT) impedance. This field must be configured prior to, or at the same time as, setting the ZCAL_ENA field.<br>2: 150ohms<br>5: 75ohms<br>8: 50ohms<br>Other values are reserved. | 0x3 |
| ZCAL_PROG | 4:1 | R/W | Together with the external reference resistor this field configures the SSTL output impedance. This field must be configured prior to, or at the same time as, setting the ZCAL_ENA field.<br>11: 40ohms<br>Other values are reserved. | 0xB |
| ZCAL_ENA | 0 | One-shot | Set this field to start automatic SSTL output and ODT impedance calibration. This field is cleared when the automatic calibration has completed. | 0x0 |

## 7.15.9 ICPU_CFG:TWI_DELAY

**Parent:** ICPU_CFG

**Instances:** 1

*Table 674 •* **Registers in TWI_DELAY**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| TWI_CONFIG | 0x00000000 | 1 | Configuration registers | Page 504 |

### 7.15.9.1 ICPU_CFG:TWI_DELAY:TWI_CONFIG

**Parent:** ICPU_CFG:TWI_DELAY

**Instances:** 1

*Table 675 •* **Fields in TWI_CONFIG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TWI_CNT_RELOAD | 8:1 | R/W | Configure the hold time delay to apply to SDA after SCK when transmitting from the device. The delay depends on the VCore system clock period. If for example the VCore system clock is 125MHz then the period is 8ns, in turn the hold time will then be (TWI_CNT_RELOAD+2) * 8ns. Replace the clock period for other VCore system frequencies. The resulting value should be as close to 300ns as possible without going below 300ns. | 0x00 |
| TWI_DELAY_ENABLE | 0 | R/W | Set this field to enable hold time on the TWI SDA output. When enabled the TWI_CONFIG.TWI_CNT_RELOAD field determines the amount of hold time to apply to SDA. | 0x0 |

# 7.16 UART

*Table 676 •* **Register Groups in UART**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| UART | 0x00000000 | 1 | UART registers | Page 505 |

## 7.16.1 UART:UART

**Parent:** UART

**Instances:** 1

*Table 677 •* **Registers in UART**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| RBR_THR | 0x00000000 | 1 | Receive Buffer/Transmit Holding Register/Divisor (Low) | Page 506 |
| IER | 0x00000004 | 1 | Interrupt Enable Register/Divisor (High) | Page 507 |

*Table 677 •* **Registers in UART *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| IIR_FCR | 0x00000008 | 1 | Interrupt Identification Register/FIFO Control Register | Page 508 |
| LCR | 0x0000000C | 1 | Line Control Register | Page 510 |
| MCR | 0x00000010 | 1 | Modem Control Register | Page 511 |
| LSR | 0x00000014 | 1 | Line Status Register | Page 512 |
| MSR | 0x00000018 | 1 | Modem Status Register | Page 515 |
| SCR | 0x0000001C | 1 | Scratchpad Register | Page 516 |
| USR | 0x0000007C | 1 | UART Status Register | Page 516 |

### 7.16.1.1 UART:UART:RBR_THR

**Parent:** UART:UART

**Instances:** 1

When the LCR.DLAB is set, this register is the lower 8 bits of the 16-bit Divisor register that contains the baud rate divisor for the UART.

The output baud rate is equal to the VCore system clock frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (VCore clock freq)/(16 * divisor). Note that with the Divisor set to zero, the baud clock is disabled and no serial communications occur. In addition, once this register is set, wait at least 0.1us before transmitting or receiving data.

*Table 678 •* **Fields in RBR_THR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RBR_THR | 7:0 | R/W | Use this register to access the Rx and Tx FIFOs.<br>When reading: The data in this register is valid only if LSR.DR is set. If FIFOs are disabled (IIR_FCR.FIFOE), the data in this register must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. When FIFOs are enabled (IIR_FCR.FIFOE), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data is lost and an overrun error occurs.<br>When writing: Data should only be written to this register when the LSR.THRE indicates that there is room in the FIFO. If FIFOs are disabled (IIR_FCR.FIFOE), writes to this register while LSR.THRE is zero, causes the register to be overwritten. When FIFOs are enabled (IIR_FCR.FIFOE) and LSR.THRE is set, 16 characters may be written to this register before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost. | 0x00 |

### 7.16.1.2 UART:UART:IER

**Parent:** UART:UART

**Instances:** 1

When the LCR.DLAB is set, this register is the upper 8 bits of the 16-bit Divisor register that contains the baud rate divisor for the UART. For more information and a description of how to calculate the baud rate, see RBR_THR.

*Table 679 •* **Fields in IER**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PTIME | 7 | R/W | Programmable THRE interrupt mode enable. This is used to enable or disable the generation of THRE interrupt.<br>0: Disabled<br>1: Enabled | 0x0 |
| EDSSI | 3 | R/W | Enable modem status interrupt. This is used to enable or disable the generation of Modem Status interrupt. This is the fourth highest priority interrupt.<br>0: Disabled<br>1: Enabled | 0x0 |
| ELSI | 2 | R/W | Enable receiver line status interrupt. This is used to enable or disable the generation of Receiver Line Status interrupt. This is the highest priority interrupt.<br>0: Disabled<br>1: Enabled | 0x0 |
| ETBEI | 1 | R/W | Enable transmit holding register empty interrupt. This is used to enable or disable the generation of Transmitter Holding Register Empty interrupt. This is the third highest priority interrupt.<br>0: Disabled<br>1: Enabled | 0x0 |
| ERBFI | 0 | R/W | Enable received data available interrupt. This is used to enable or disable the generation of Received Data Available interrupt and the Character Timeout interrupt (if FIFOs are enabled). These are the second highest priority interrupts.<br>0: Disabled<br>1: Enabled | 0x0 |

### 7.16.1.3 UART:UART:IIR_FCR

**Parent:** UART:UART

**Instances:** 1

This register has special meaning when reading, here the lowest 4 bits indicate interrupting sources. The encoding is as follows:

0110; type: Receiver line status, priority: Highest. Overrun/parity/ framing errors or break interrupt. Cleared by reading LSR.

0100; type: Received data available, priority: Second. RCVR FIFO trigger level reached. Cleared when FIFO drops below the trigger level.

1100; type: Character timeout indication, priority: Second. No characters in or out of the RCVR FIFO during the last four character times and there is at least 1 character in it during this time. Cleared by reading the receiver buffer register.

0010; type: Transmit holding register empty, priority: Third. Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled). Cleared by reading the IIR register (if source of interrupt); or, writing into THR (THRE Mode disabled) or XMIT FIFO above threshold (THRE Mode enabled).

0000; type: Modem status, priority: Fourth. Clear to send. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt. Cleared by reading the Modem status register.

0111; type: Busy detect indication, priority: Fifth. Master has tried to write to the Line Control register while the UART is busy (USR[0] is set to one). Cleared by reading the UART status register.

0001: No interrupting sources.

***Table 680 •*** **Fields in IIR_FCR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FIFOSE_RT | 7:6 | R/W | When reading this field, the current status of the FIFO is returned; 00 for disabled or 11 for enabled. Writing this field selects the trigger level in the receive FIFO at which the Received Data Available interrupt is generated (see encoding.) In auto flow control mode, it is used to determine when to generate back-pressure using the RTS signal.<br>00: 1 character in the Rx FIFO<br>01: Rx FIFO 1/4 full<br>10: Rx FIFO 1/2 full<br>11: Rx FIFO 2 less than full | 0x1 |
| TET | 5:4 | R/W | Tx empty trigger. When the THRE mode is enabled (IER.PTIME), this field selects the empty threshold level at which the THRE Interrupts are generated.<br>00: Tx FIFO empty<br>01: 2 characters in the Tx FIFO<br>10: Tx FIFO 1/4 full<br>11: Tx FIFO 1/2 full | 0x0 |
| XFIFOR | 2 | R/W | This description is valid for writes only. Reading this field has special meaning; for more information, see the general register description.<br>Tx FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is self-clearing. It is not necessary to clear this bit. | 0x0 |

*Table 680 •* **Fields in IIR_FCR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RFIFOR | 1 | R/W | This description is valid for writes only. Reading this field has special meaning; for more information, see the general register description.<br>Rx FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is self-clearing. It is not necessary to clear this bit. | 0x0 |
| FIFOE | 0 | R/W | This description is valid for writes only. Reading this field has special meaning; for more information, see the general register description.<br>FIFO Enable. This enables or disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed, both the XMIT and RCVR controller portion of FIFOs are reset. | 0x0 |

### 7.16.1.4 UART:UART:LCR

**Parent:** UART:UART

**Instances:** 1

Writes can be made to this register, with the exception of the BC field, only when UART is not busy, that is, when USR.BUSY is zero. This register can always be read.

*Table 681 •* **Fields in LCR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DLAB | 7 | R/W | Divisor latch access bit. This bit is used to enable reading and writing of the Divisor registers (RBR_THR and IER) to set the baud rate of the UART. To access other registers, this bit must be cleared after initial baud rate setup. | 0x0 |
| BC | 6 | R/W | Break control bit.This bit is used to cause a break condition to be transmitted to the receiving device. If set to one, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the serial output is forced low until the Break bit is cleared. | 0x0 |

*Table 681 •* **Fields in LCR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EPS | 4 | R/W | Even parity select. This bit is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked. | 0x0 |
| PEN | 3 | R/W | Parity enable. This bit is used to enable or disable parity generation and detection in both transmitted and received serial characters.<br>0: Parity disabled<br>1: Parity enabled | 0x0 |
| STOP | 2 | R/W | Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.<br>If set to one and the data bits are set to 5 (LCR.DLS), one and a half stop bits are transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit.<br>0: 1 stop bit<br>1: 1.5 stop bits when LCR.DLS is zero, otherwise, 2 stop bits | 0x0 |
| DLS | 1:0 | R/W | Data length select. This is used to select the number of data bits per character that the peripheral transmits and receives. The following settings specify the number of bits that may be selected.<br>00: 5 bits<br>01: 6 bits<br>10: 7 bits<br>11: 8 bits | 0x0 |

### 7.16.1.5 UART:UART:MCR

**Parent:** UART:UART

**Instances:** 1

*Table 682 •* **Fields in MCR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AFCE | 5 | R/W | Auto flow control enable. This mode requires that FIFOs are enabled and that MCR.RTS is set.<br>0: Auto flow control mode disabled<br>1: Auto flow control mode enabled | 0x0 |
| LB | 4 | R/W | Loopback Bit. This is used to put the UART into a diagnostic mode for test purposes.<br>The transmit line is held high, while serial transmit data is looped back to the receive line internally. In this mode, all the interrupts are fully functional. In addition, in loopback mode, the modem control input CTS is disconnected, and the modem control output RTS is looped back to the input internally. | 0x0 |
| RTS | 1 | R/W | Request to send. This is used to directly control the Request to Send (RTS) output. The RTS output is used to inform the partner that the UART is ready to exchange data.<br>The RTS is still controlled from this field when Auto RTS Flow Control is enabled (MCR.AFCE), but the output can be forced high by the flow control mechanism. If this field is cleared, the UART permanently indicates backpressure to the partner.<br>0: RTS is set high<br>1: RTS is set low | 0x0 |

### 7.16.1.6 UART:UART:LSR

**Parent:** UART:UART

**Instances:** 1

*Table 683 •* **Fields in LSR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RFE | 7 | R/W | Receiver FIFO error bit. This bit is only valid when FIFOs are enabled. This is used to indicate whether there is at least one parity error, framing error, or break indication in the FIFO.<br>This bit is cleared when the LSR is read, the character with the error is at the top of the receiver FIFO, and there are no subsequent errors in the FIFO.<br>0: No error in Rx FIFO<br>1: Error in Rx FIFO | 0x0 |
| TEMT | 6 | R/W | Transmitter empty bit. If FIFOs are enabled, this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. | 0x1 |
| THRE | 5 | R/W | If FIFO (IIR_FCR.FIFOE) and THRE mode are enabled (IER.PTIME), this bit indicates that the Tx FIFO is full. Otherwise, this bit indicates that the Tx FIFO is empty. | 0x1 |
| BI | 4 | R/W | Break interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. It is set whenever the serial input is held in a logic 0 state for longer than the sum of start time + data bits + parity + stop bits.<br>A break condition on serial input causes one and only one character, consisting of all-zeros, to be received by the UART.<br>In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit.<br>In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. | 0x0 |

*Table 683 •* **Fields in LSR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FE | 3 | R/W | Framing error bit. This is used to indicate the a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.<br>A framing error is associated with a received character. Therefore, in FIFO mode, an error is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues to receive the other bit, that is, data and/or parity, and then stops. Note that this field is set if a break interrupt has occurred, as indicated by Break Interrupt (LSR.BI).<br>This field is cleared on read.<br>0: No framing error<br>1: Framing error | 0x0 |
| PE | 2 | R/W | Parity error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable bit (LCR.PEN) is set.<br>A parity error is associated with a received character. Therefore, in FIFO mode, an error is revealed when the character with the parity error arrives at the top of the FIFO. Note that this field is set if a break interrupt has occurred, as indicated by Break Interrupt (LSR.BI).<br>This field is cleared on read.<br>0: No parity error<br>1: Parity error | 0x0 |

*Table 683 •* **Fields in LSR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OE | 1 | R/W | Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In non-FIFO mode, the OE bit is set when a new character arrives before the previous character was read. When this happens, the data in the RBR is overwritten.<br>In FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost.<br>This field is cleared on read.<br>0: No overrun error<br>1: Overrun error | 0x0 |
| DR | 0 | R/W | Data ready. This is used to indicate that the receiver contains at least one character in the receiver FIFO. This bit is cleared when the RX FIFO is empty.<br>0: No data ready<br>1: Data ready | 0x0 |

### 7.16.1.7 UART:UART:MSR

**Parent:** UART:UART

**Instances:** 1

*Table 684 •* **Fields in MSR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CTS | 4 | R/O | Clear to send. This field indicates the current state of the modem control line, CTS. When the Clear to Send input (CTS) is asserted, it is an indication that the partner is ready to exchange data with the UART.<br>0: CTS input is deasserted (logic 0)<br>1: CTS input is asserted (logic 1) | 0x0 |

*Table 684 •* **Fields in MSR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DCTS | 0 | R/O | Delta clear to send. This is used to indicate that the modem control line, CTS, has changed since the last time the MSR was read. Reading the MSR clears the DCTS bit.<br>Note: If the DCTS bit is not set, the CTS signal is asserted, and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed, if the CTS signal remains asserted. A read of the MSR after reset can be performed to prevent unwanted interrupts.<br>0: No change on CTS since the last read of the MSR<br>1: Change on CTS since the last read of the MSR | 0x0 |

### 7.16.1.8 UART:UART:SCR

**Parent:** UART:UART

**Instances:** 1

*Table 685 •* **Fields in SCR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SCR | 7:0 | R/W | This register is for programmers to use as a temporary storage space. It has no functional purpose for the UART. | 0x00 |

### 7.16.1.9 UART:UART:USR

**Parent:** UART:UART

**Instances:** 1

*Table 686 •* **Fields in USR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BUSY | 0 | R/O | UART busy.<br>0: UART is idle or inactive<br>1: UART is busy (actively transferring data) | 0x0 |

# 7.17 TWI

*Table 687 •* **Register Groups in TWI**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| TWI | 0x00000000 | 1 | Two-Wire Interface Controller Registers | Page 517 |

## 7.17.1 TWI:TWI

**Parent:** TWI

**Instances:** 1

*Table 688 •* **Registers in TWI**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CFG | 0x00000000 | 1 | TWI Configuration | Page 518 |
| TAR | 0x00000004 | 1 | Target Address | Page 520 |
| SAR | 0x00000008 | 1 | Slave Address | Page 520 |
| DATA_CMD | 0x00000010 | 1 | Rx/Tx Data Buffer and Command | Page 521 |
| SS_SCL_HCNT | 0x00000014 | 1 | Standard Speed TWI Clock SCL High Count | Page 522 |
| SS_SCL_LCNT | 0x00000018 | 1 | Standard Speed TWI Clock SCL Low Count | Page 523 |
| FS_SCL_HCNT | 0x0000001C | 1 | Fast Speed TWI Clock SCL High Count | Page 523 |
| FS_SCL_LCNT | 0x00000020 | 1 | Fast Speed TWI Clock SCL Low Count | Page 524 |
| INTR_STAT | 0x0000002C | 1 | Interrupt Status | Page 524 |
| INTR_MASK | 0x00000030 | 1 | Interrupt Mask | Page 524 |
| RAW_INTR_STAT | 0x00000034 | 1 | Raw Interrupt Status | Page 525 |
| RX_TL | 0x00000038 | 1 | Receive FIFO Threshold | Page 529 |
| TX_TL | 0x0000003C | 1 | Transmit FIFO Threshold | Page 530 |
| CLR_INTR | 0x00000040 | 1 | Clear Combined and Individual Interrupt | Page 530 |
| CLR_RX_UNDER | 0x00000044 | 1 | Clear RX_UNDER Interrupt | Page 530 |
| CLR_RX_OVER | 0x00000048 | 1 | Clear RX_OVER Interrupt | Page 531 |
| CLR_TX_OVER | 0x0000004C | 1 | Clear TX_OVER Interrupt | Page 531 |
| CLR_RD_REQ | 0x00000050 | 1 | Clear RD_REQ Interrupt | Page 531 |
| CLR_TX_ABRT | 0x00000054 | 1 | Clear TX_ABRT Interrupt | Page 531 |

*Table 688 •* **Registers in TWI *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CLR_RX_DONE | 0x00000058 | 1 | Clear RX_DONE Interrupt | Page 532 |
| CLR_ACTIVITY | 0x0000005C | 1 | Clear ACTIVITY Interrupt | Page 532 |
| CLR_STOP_DET | 0x00000060 | 1 | Clear STOP_DET Interrupt | Page 532 |
| CLR_START_DET | 0x00000064 | 1 | Clear START_DET Interrupt | Page 533 |
| CLR_GEN_CALL | 0x00000068 | 1 | Clear GEN_CALL Interrupt | Page 533 |
| CTRL | 0x0000006C | 1 | TWI Control | Page 533 |
| STAT | 0x00000070 | 1 | TWI Status | Page 534 |
| TXFLR | 0x00000074 | 1 | Transmit FIFO Level | Page 535 |
| RXFLR | 0x00000078 | 1 | Receive FIFO Level | Page 536 |
| TX_ABRT_SOURCE | 0x00000080 | 1 | Transmit Abort Source | Page 536 |
| SDA_SETUP | 0x00000094 | 1 | SDA Setup | Page 538 |
| ACK_GEN_CALL | 0x00000098 | 1 | ACK General Call | Page 538 |
| ENABLE_STATUS | 0x0000009C | 1 | Enable Status | Page 539 |

### 7.17.1.1 TWI:TWI:CFG

**Parent:** TWI:TWI

**Instances:** 1

*Table 689 •* **Fields in CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SLAVE_DIS | 6 | R/W | This bit controls whether the TWI controller has its slave disabled. If this bit is set (slave is disabled), the controller functions only as a master and does not perform any action that requires a slave.<br>'0': slave is enabled<br>'1': slave is disabled | 0x1 |

*Table 689 •* **Fields in CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESTART_ENA | 5 | R/W | Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several operations.<br>When RESTART is disabled, the master is prohibited from performing the following functions:<br>* Change direction within a transfer (split)<br>* Send a START BYTE<br>* Combined format transfers in 7-bit addressing modes<br>* Read operation with a 10-bit address<br>* Send multiple bytes per transfer<br>By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple transfers. If the above operations are performed, it will result in setting RAW_INTR_STAT.TX_ABRT.<br>'0': disable<br>'1': enable | 0x1 |
| MASTER_10BITADDR | 4 | R/W | Controls whether transfers starts in 7- or 10-bit addressing mode when acting as a master.<br>'0': 7-bit addressing<br>'1': 10-bit addressing | 0x0 |
| SLAVE_10BITADDR | 3 | R/W | Controls whether the TWI controller responds to 7- or 10-bit addresses in slave mode. In 7-bit mode; transactions that involve 10-bit addressing are ignored and only the lower 7 bits of the SAR register are compared.<br>'0': 7-bit addressing.<br>'1': 10-bit addressing. | 0x0 |
| SPEED | 2:1 | R/W | These bits control at which speed the TWI controller operates; its setting is relevant only in master mode. Hardware protects against illegal values being programmed by software.<br>'1': standard mode (100 kbit/s)<br>'2': fast mode (400 kbit/s) | 0x2 |

*Table 689 •* **Fields in CFG *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MASTER_ENA | 0 | R/W | This bit controls whether the TWI master is enabled.<br>'0': master disabled<br>'1': master enabled | 0x1 |

### 7.17.1.2 TWI:TWI:TAR

**Parent:** TWI:TWI

**Instances:** 1

*Table 690 •* **Fields in TAR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GC_OR_START_ENA | 11 | R/W | This bit indicates whether software performs a General Call or START BYTE command.<br>'0': ignore bit 10 GC_OR_START and use TAR normally<br>'1': perform special TWI command as specified in GC_OR_START bit | 0x0 |
| GC_OR_START | 10 | R/W | If TAR.SPECIAL is set to 1, then this bit indicates whether a General Call or START byte command is to be performed.<br>'0': General Call Address - after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting RAW_INTR_STAT.TX_ABRT. The TWI controller remains in General Call mode until the TAR.SPECIAL field is cleared.<br>'1': START BYTE | 0x0 |
| TAR | 9:0 | R/W | This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits.<br>If the TAR and SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave. | 0x055 |

### 7.17.1.3 TWI:TWI:SAR

**Parent:** TWI:TWI

**Instances:** 1

*Table 691* • **Fields in SAR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SAR | 9:0 | R/W | The SAR holds the slave address when the TWI is operating as a slave. For 7-bit addressing, only SAR[6:0] is used.<br>This register can be written only when the TWI interface is disabled (ENABLE = 0). | 0x055 |

### 7.17.1.4 TWI:TWI:DATA_CMD

**Parent:** TWI:TWI

**Instances:** 1

*Table 692 •* **Fields in DATA_CMD**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CMD | 8 | R/W | This bit controls whether a read or a write is performed. This bit does not control the direction when the TWI acts as a slave. It controls only the direction when it acts as a master.<br>When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a "don't care" because writes to this register are not required. In slave-transmitter mode, a "0" indicates that CPU data is to be transmitted and as DATA.<br>When programming this bit, please remember the following:<br>attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (RAW_INTR_STAT.R_TX_ABRT), unless TAR.SPECIAL has been cleared.<br>If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.<br>NOTE: It is possible that while attempting a master TWI read transfer, a RD_REQ interrupt may have occurred simultaneously due to a remote TWI master addressing this controller. In this type of scenario, the TWI controller ignores the DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt.<br>'1' = Read<br>'0' = Write | 0x0 |
| DATA | 7:0 | R/W | This register contains the data to be transmitted or received on the TWI bus. If you are writing to this register and want to perform a read, this field is ignored by the controller. However, when you read this register, these bits return the value of data received on the TWI interface. | 0x00 |

### 7.17.1.5 TWI:TWI:SS_SCL_HCNT

**Parent:** TWI:TWI

**Instances:** 1

The clock for the TWI controller is the VCore system clock. This field must be set accordingly to the VCore system frequency; value = (4us/VCore clock period) - 8.

Example: a 178.6MHz clock correspond to a period of 5.6ns, for this frequency this field must not be set lower than (round up): 707 = (4us/5.6ns) - 8.

***Table 693 •*** **Fields in SS_SCL_HCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SS_SCL_HCNT | 15:0 | R/W | This register sets the SCL clock divider for the high-period in standard speed. This value must result in a high period of no less than 4us. | 0x033A |

### 7.17.1.6 TWI:TWI:SS_SCL_LCNT

**Parent:** TWI:TWI

**Instances:** 1

The clock for the TWI controller is the VCore system clock. This field must be set accordingly to the VCore system frequency; value = (4.7us/VCore clock period) - 1.

Example: a 178.6MHz clock correspond to a period of 5.6ns, for this frequency this field must not be set lower than (round up): 839 = (4.7us/5.6ns) - 1.

***Table 694 •*** **Fields in SS_SCL_LCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SS_SCL_LCNT | 15:0 | R/W | This register sets the SCL clock divider for the low-period in standard speed. This value must result in a value no less than 4.7us. | 0x03D3 |

### 7.17.1.7 TWI:TWI:FS_SCL_HCNT

**Parent:** TWI:TWI

**Instances:** 1

The clock for the TWI controller is the VCore system clock. This field must be set accordingly to the VCore system frequency; value = (0.6us/VCore clock period) - 8.

Example: a 178.6MHz clock correspond to a period of 5.6ns, for this frequency this field must not be set lower than (round up): 100 = (0.6us/5.6ns) - 8.

***Table 695 •*** **Fields in FS_SCL_HCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FS_SCL_HCNT | 15:0 | R/W | This register sets the SCL clock divider for the high-period in fast speed. This value must result in a value no less than 0.6us. | 0x0075 |

### 7.17.1.8 TWI:TWI:FS_SCL_LCNT

**Parent:** TWI:TWI

**Instances:** 1

The clock for the TWI controller is the VCore system clock. This field must be set accordingly to the VCore system frequency; value = (1.3us/VCore clock period) - 1.

Example: a 178.6MHz clock correspond to a period of 5.6ns, for this frequency this field must not be set lower than (round up): 232 = (1.3us/5.6ns) - 1.

***Table 696 •*** **Fields in FS_SCL_LCNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FS_SCL_LCNT | 15:0 | R/W | This register sets the SCL clock divider for the low-period in fast speed. This value must result in a value no less than 1.3us. | 0x010E |

### 7.17.1.9 TWI:TWI:INTR_STAT

**Parent:** TWI:TWI

**Instances:** 1

Each field in this register has a corresponding mask field in the INTR_MASK register. These fields are cleared by reading the matching interrupt clear register. The unmasked raw versions of these fields are available in the RAW_INTR_STAT register.

See RAW_INTR_STAT for a description of these fields

***Table 697 •*** **Fields in INTR_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| GEN_CALL | 11 | R/O | | 0x0 |
| START_DET | 10 | R/O | | 0x0 |
| STOP_DET | 9 | R/O | | 0x0 |
| ACTIVITY | 8 | R/O | | 0x0 |
| RX_DONE | 7 | R/O | | 0x0 |
| TX_ABRT | 6 | R/O | | 0x0 |
| RD_REQ | 5 | R/O | | 0x0 |
| TX_EMPTY | 4 | R/O | | 0x0 |
| TX_OVER | 3 | R/O | | 0x0 |
| RX_FULL | 2 | R/O | | 0x0 |
| RX_OVER | 1 | R/O | | 0x0 |
| RX_UNDER | 0 | R/O | | 0x0 |

### 7.17.1.10 TWI:TWI:INTR_MASK

**Parent:** TWI:TWI

**Instances:** 1

These fields mask the corresponding interrupt status fields (RAW_INTR_STAT). They are active high; a value of 0 prevents the corresponding field in RAW_INTR_STAT from generating an interrupt.

***Table 698 •*** **Fields in INTR_MASK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| M_GEN_CALL | 11 | R/W | | 0x1 |
| M_START_DET | 10 | R/W | | 0x0 |
| M_STOP_DET | 9 | R/W | | 0x0 |
| M_ACTIVITY | 8 | R/W | | 0x0 |
| M_RX_DONE | 7 | R/W | | 0x1 |
| M_TX_ABRT | 6 | R/W | | 0x1 |
| M_RD_REQ | 5 | R/W | | 0x1 |
| M_TX_EMPTY | 4 | R/W | | 0x1 |
| M_TX_OVER | 3 | R/W | | 0x1 |
| M_RX_FULL | 2 | R/W | | 0x1 |
| M_RX_OVER | 1 | R/W | | 0x1 |
| M_RX_UNDER | 0 | R/W | | 0x1 |

### 7.17.1.11 TWI:TWI:RAW_INTR_STAT

**Parent:** TWI:TWI

**Instances:** 1

Unlike the INTR_STAT register, these fields are not masked so they always show the true status of the TWI controller.

***Table 699 •*** **Fields in RAW_INTR_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| R_GEN_CALL | 11 | R/O | Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling TWI controller or when the CPU reads bit 0 of the CLR_GEN_CALL register. The TWI controller stores the received data in the Rx buffer. | 0x0 |
| R_START_DET | 10 | R/O | Indicates whether a START or RESTART condition has occurred on the TWI regardless of whether the TWI controller is operating in slave or master mode. | 0x0 |
| R_STOP_DET | 9 | R/O | Indicates whether a STOP condition has occurred on the TWI controller regardless of whether the TWI controller is operating in slave or master mode. | 0x0 |

*Table 699 •* **Fields in RAW_INTR_STAT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| R_ACTIVITY | 8 | R/O | This bit captures TWI activity and stays set until it is cleared. There are four ways to clear it:<br>* Disabling the TWI controller<br>* Reading the CLR_ACTIVITY register<br>* Reading the CLR_INTR register<br>* VCore system reset<br>Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the TWI controller module is idle, this bit remains set until cleared, indicating that there was activity on the bus. | 0x0 |
| R_RX_DONE | 7 | R/O | When the TWI controller is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done. | 0x0 |

*Table 699 •* **Fields in RAW_INTR_STAT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| R_TX_ABRT | 6 | R/O | This bit is set to 1 when the TWI controller is acting as a master is unable to complete a command that the processor has sent. The conditions that set this field are:<br>* No slave acknowledges the address byte.<br>* The addressed slave receiver does not acknowledge a byte of data.<br>* Attempting to send a master command when configured only to be a slave.<br>* When CFG.RESTART_ENA is set to 0 (RESTART condition disabled), and the processor attempts to issue a TWI function that is impossible to perform without using RESTART conditions.<br>* High-speed master code is acknowledged (this controller does not support high-speed).<br>* START BYTE is acknowledged.<br>* General Call address is not acknowledged.<br>* When a read request interrupt occurs and the processor has previously placed data in the Tx buffer that has not been transmitted yet. This data could have been intended to service a multi-byte RD_REQ that ended up having fewer numbers of bytes requested.<br>*The TWI controller loses arbitration of the bus between transfers and is then accessed as a slave-transmitter.<br>* If a read command is issued after a General Call command has been issued. Disabling the TWI reverts it back to normal operation.<br>* If the CPU attempts to issue read command before a RD_REQ is serviced.<br>Anytime this bit is set, the contents of the transmit and receive buffers are flushed. | 0x0 |

*Table 699 •* **Fields in RAW_INTR_STAT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| R_RD_REQ | 5 | R/O | This bit is set to 1 when the TWI controller acts as a slave and another TWI master is attempting to read data from this controller. The TWI controller holds the TWI bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the DATA_CMD register. This bit is set to 0 just after the required data is written to the DATA_CMD register. | 0x0 |
| R_TX_EMPTY | 4 | R/O | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When ENABLE is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ENABLE_STATUS.BUSY=0, this bit is set to 0. | 0x0 |
| R_TX_OVER | 3 | R/O | Set during transmit if the transmit buffer is filled to TX_BUFFER_DEPTH and the processor attempts to issue another TWI command by writing to the DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ENABLE_STATUS.BUSY goes to 0, this interrupt is cleared. | 0x0 |

*Table 699 •* **Fields in RAW_INTR_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| R_RX_FULL | 2 | R/O | Set when the receive buffer reaches or goes above the RX_TL threshold in the RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (ENABLE=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the ENABLE field is programmed with a 0, regardless of the activity that continues. | 0x0 |
| R_RX_OVER | 1 | R/O | Set if the receive buffer is completely filled to RX_BUFFER_DEPTH and an additional byte is received from an external TWI device. The TWI controller acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (ENABLE=0), this bit keeps its level until the master or slave state machines go into idle, and when ENABLE_STATUS.BUSY goes to 0, this interrupt is cleared. | 0x0 |
| R_RX_UNDER | 0 | R/O | Set if the processor attempts to read the receive buffer when it is empty by reading from the DATA_CMD register. If the module is disabled (ENABLE=0), this bit keeps its level until the master or slave state machines go into idle, and when ENABLE_STATUS.BUSY goes to 0, this interrupt is cleared. | 0x0 |

### 7.17.1.12 TWI:TWI:RX_TL

**Parent:** TWI:TWI

**Instances:** 1

*Table 700* • **Fields in RX_TL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_TL | 2:0 | R/W | Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in RAW_INTR_STAT register). The valid range is 0-7. A value of 0 sets the threshold for 1 entry, and a value of 7 sets the threshold for 8 entries. | 0x0 |

### 7.17.1.13 TWI:TWI:TX_TL

**Parent:** TWI:TWI

**Instances:** 1

*Table 701* • **Fields in TX_TL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_TL | 2:0 | R/W | Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in RAW_INTR_STAT register). The valid range is 0-7. A value of 0 sets the threshold for 0 entries, and a value of 7 sets the threshold for 7 entries. | 0x0 |

### 7.17.1.14 TWI:TWI:CLR_INTR

**Parent:** TWI:TWI

**Instances:** 1

*Table 702* • **Fields in CLR_INTR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_INTR | 0 | R/O | Read this register to clear the combined interrupt, all individual interrupts, and the TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the TX_ABRT_SOURCE register for an exception to clearing TX_ABRT_SOURCE. | 0x0 |

### 7.17.1.15 TWI:TWI:CLR_RX_UNDER

**Parent:** TWI:TWI

**Instances:** 1

*Table 703* • **Fields in CLR_RX_UNDER**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_RX_UNDER | 0 | R/O | Read this register to clear the R_RX_UNDER interrupt (bit 0) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.16 TWI:TWI:CLR_RX_OVER

**Parent:** TWI:TWI

**Instances:** 1

*Table 704* • **Fields in CLR_RX_OVER**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_RX_OVER | 0 | R/O | Read this register to clear the R_RX_OVER interrupt (bit 1) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.17 TWI:TWI:CLR_TX_OVER

**Parent:** TWI:TWI

**Instances:** 1

*Table 705* • **Fields in CLR_TX_OVER**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_TX_OVER | 0 | R/O | Read this register to clear the R_TX_OVER interrupt (bit 3) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.18 TWI:TWI:CLR_RD_REQ

**Parent:** TWI:TWI

**Instances:** 1

*Table 706* • **Fields in CLR_RD_REQ**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_RD_REQ | 0 | R/O | Read this register to clear the R_RD_REQ interrupt (bit 5) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.19 TWI:TWI:CLR_TX_ABRT

**Parent:** TWI:TWI

**Instances:** 1

*Table 707* • **Fields in CLR_TX_ABRT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_TX_ABRT | 0 | R/O | Read this register to clear the R_TX_ABRT interrupt (bit 6) of the RAW_INTR_STAT register, and the TX_ABRT_SOURCE register. Refer to Bit 9 of the TX_ABRT_SOURCE register for an exception to clearing TX_ABRT_SOURCE. | 0x0 |

### 7.17.1.20 TWI:TWI:CLR_RX_DONE

**Parent:** TWI:TWI

**Instances:** 1

*Table 708* • **Fields in CLR_RX_DONE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_RX_DONE | 0 | R/O | Read this register to clear the R_RX_DONE interrupt (bit 7) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.21 TWI:TWI:CLR_ACTIVITY

**Parent:** TWI:TWI

**Instances:** 1

*Table 709* • **Fields in CLR_ACTIVITY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_ACTIVITY | 0 | R/O | Reading this register clears the ACTIVITY interrupt if the TWI controller is not active anymore. If the TWI controller is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the R_ACTIVITY interrupt (bit 8) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.22 TWI:TWI:CLR_STOP_DET

**Parent:** TWI:TWI

**Instances:** 1

*Table 710* • **Fields in CLR_STOP_DET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_STOP_DET | 0 | R/O | Read this register to clear the R_STOP_DET interrupt (bit 9) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.23 TWI:TWI:CLR_START_DET

**Parent:** TWI:TWI

**Instances:** 1

*Table 711* • **Fields in CLR_START_DET**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_START_DET | 0 | R/O | Read this register to clear the R_START_DET interrupt (bit 10) of the RAW_INTR_STAT register. | 0x0 |

### 7.17.1.24 TWI:TWI:CLR_GEN_CALL

**Parent:** TWI:TWI

**Instances:** 1

*Table 712* • **Fields in CLR_GEN_CALL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLR_GEN_CALL | 0 | R/O | Read this register to clear the R_GEN_CALL interrupt (bit 11) of RAW_INTR_STAT register. | 0x0 |

### 7.17.1.25 TWI:TWI:CTRL

**Parent:** TWI:TWI

**Instances:** 1

*Table 713 •* **Fields in CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ENABLE | 0 | R/W | Controls whether the TWI controller is enabled. Software can disable the controller while it is active. However, it is important that care be taken to ensure that the controller is disabled properly.<br>When TWI controller is disabled, the following occurs:<br>* The TX FIFO and RX FIFO get flushed.<br>* The interrupt bits in the RAW_INTR_STAT register are cleared.<br>* Status bits in the INTR_STAT register are still active until the TWI controller goes into IDLE state.<br>If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the controller stops the current transfer at the end of the current byte and does not acknowledge the transfer.<br>'0': Disables TWI controller<br>'1': Enables TWI controller | 0x0 |

### 7.17.1.26 TWI:TWI:STAT

**Parent:** TWI:TWI

**Instances:** 1

*Table 714 •* **Fields in STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SLV_ACTIVITY | 6 | R/O | Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set.<br>'0': Slave FSM is in IDLE state so the Slave part of the controller is not Active<br>'1': Slave FSM is not in IDLE state so the Slave part of the controller is Active | 0x0 |

*Table 714 •* **Fields in STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MST_ACTIVITY | 5 | R/O | Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set.<br>'0': Master FSM is in IDLE state so the Master part of the controller is not Active<br>'1': Master FSM is not in IDLE state so the Master part of the controller is Active | 0x0 |
| RFF | 4 | R/O | Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.<br>'0': Receive FIFO is not full<br>'1': Receive FIFO is full | 0x0 |
| RFNE | 3 | R/O | Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO.<br>'0': Receive FIFO is empty<br>'1': Receive FIFO is not empty | 0x0 |
| TFE | 2 | R/O | Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.<br>'0': Transmit FIFO is not empty<br>'1': Transmit FIFO is empty | 0x1 |
| TFNF | 1 | R/O | Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.<br>'0': Transmit FIFO is full<br>'1': Transmit FIFO is not full | 0x1 |
| BUS_ACTIVITY | 0 | R/O | TWI Activity Status. | 0x0 |

### 7.17.1.27 TWI:TWI:TXFLR

**Parent:** TWI:TWI

**Instances:** 1

*Table 715 •* **Fields in TXFLR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TXFLR | 2:0 | R/O | Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. | 0x0 |

### 7.17.1.28 TWI:TWI:RXFLR

**Parent:** TWI:TWI

**Instances:** 1

*Table 716 •* **Fields in RXFLR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RXFLR | 2:0 | R/O | Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. | 0x0 |

### 7.17.1.29 TWI:TWI:TX_ABRT_SOURCE

**Parent:** TWI:TWI

**Instances:** 1

*Table 717 •* **Fields in TX_ABRT_SOURCE**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ABRT_SLVRD_INTX | 15 | R/W | When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 to DATA_CMD.CMD. | 0x0 |
| ABRT_SLV_ARBLOST | 14 | R/W | Slave lost the bus while transmitting data to a remote master. TX_ABRT_SOURCE[12] is set at the same time.<br>Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then the TWI controller no longer own the bus. | 0x0 |
| ABRT_SLVFLUSH_TXFIFO | 13 | R/W | Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO. | 0x0 |

*Table 717 •* **Fields in TX_ABRT_SOURCE *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ARB_LOST | 12 | R/W | Master has lost arbitration, or if TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration.<br>Note: the TWI controller can be both master and slave at the same time. | 0x0 |
| ABRT_MASTER_DIS | 11 | R/W | User tries to initiate a Master operation with the Master mode disabled. | 0x0 |
| ABRT_10B_RD_NORSTRT | 10 | R/W | The restart is disabled (RESTART_ENA bit (CFG[5]) = 0) and the master sends a read command in 10-bit addressing mode. | 0x0 |
| ABRT_SBYTE_NORSTRT | 9 | R/W | To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (CFG[5]=1), the SPECIAL bit must be cleared (TAR[11]), or the GC_OR_START bit must be cleared (TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted.<br>'1': The restart is disabled (RESTART_ENA bit (CFG[5]) = 0) and the user is trying to send a START Byte. | 0x0 |
| ABRT_SBYTE_ACKDET | 7 | R/W | Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). | 0x0 |
| ABRT_GCALL_READ | 5 | R/W | TWI controller in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (DATA_CMD[9] is set to 1). | 0x0 |
| ABRT_GCALL_NOACK | 4 | R/W | TWI controller in master mode sent a General Call and no slave on the bus acknowledged the General Call. | 0x0 |

*Table 717 •* **Fields in TX_ABRT_SOURCE *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ABRT_TXDATA_NOACK | 3 | R/W | This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). | 0x0 |
| ABRT_10ADDR2_NOACK | 2 | R/W | Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. | 0x0 |
| ABRT_10ADDR1_NOACK | 1 | R/W | Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. | 0x0 |
| ABRT_7B_ADDR_NOACK | 0 | R/W | Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. | 0x0 |

### 7.17.1.30 TWI:TWI:SDA_SETUP

**Parent:** TWI:TWI

**Instances:** 1

This field must be set accordingly to the VCore system frequency; value = 100ns/VCore clock period.

Example: a 178.6MHz clock correspond to a period of 5.6ns, for this frequency and fast TWI speed this field must not be set lower than (round up): 18 = 100ns/5.6ns. For normal TWI speed this field must not be set lower than (round up): 45 = 250ns/5.6ns.

*Table 718 •* **Fields in SDA_SETUP**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SDA_SETUP | 7:0 | R/W | This register controls the amount of time delay (in terms of number of VCore clock periods) introduced in the rising edge of SCL, relative to SDA changing, when the TWI controller services a read request in a slave-receiver operation. The minimum for fast mode is 100ns, for normal mode the minimum is 250ns. | 0x15 |

### 7.17.1.31 TWI:TWI:ACK_GEN_CALL

**Parent:** TWI:TWI

**Instances:** 1

*Table 719 •* **Fields in ACK_GEN_CALL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACK_GEN_CALL | 0 | R/W | ACK General Call. When set to 1, the TWI controller responds with a ACK when it receives a General Call. Otherwise, the controller responds with a NACK. | 0x1 |

### 7.17.1.32 TWI:TWI:ENABLE_STATUS

**Parent:** TWI:TWI

**Instances:** 1

*Table 720 •* **Fields in ENABLE_STATUS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SLV_FIFO_FILLED_AND_FLUSHED | 2 | R/O | Slave FIFO Filled and Flushed. This bit indicates if a Slave-Receiver operation has been aborted with at least 1 data byte received from a TWI transfer due to the setting of ENABLE from 1 to 0.<br>When read as 1, the TWI controller is deemed to have been actively engaged in an aborted TWI transfer (with matching address) and the data phase of the TWI transfer has been entered, even though the data byte has been responded with a NACK.<br>When read as 0, the TWI controller is deemed to have been disabled when the TWI bus is idle. | 0x0 |
| SLV_RX_ABORTED | 1 | R/O | Slave-Receiver Operation Aborted. This bit indicates if a Slave-Receiver operation has been aborted due to the setting of the ENABLE register from 1 to 0.<br>When read as 1, the TWI controller is deemed to have forced a NACK during any part of a TWI transfer, irrespective of whether the TWI address matches the slave address set in the TWI controller (SAR register).<br>When read as 0, the TWI controller is deemed to have been disabled when the TWI bus is idle. | 0x0 |

*Table 720 •* **Fields in ENABLE_STATUS *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| BUSY | 0 | R/O | When read as 1, the TWI controller is deemed to be actively involved in an TWI transfer, irrespective of whether being in an address or data phase for all master or slave modes. When read as 0, the TWI controller is deemed completely inactive. | 0x0 |

# 7.18 SBA

*Table 721 •* **Register Groups in SBA**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SBA | 0x00000000 | 1 | Shared Bus arbiter registers | Page 540 |

## 7.18.1 SBA:SBA

**Parent:** SBA

**Instances:** 1

Configurations for the Shared Bus of the CPU system.

*Table 722 •* **Registers in SBA**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PL1 | 0x00000000 | 1 | Arbitration Priority CPU | Page 540 |
| PL2 | 0x00000004 | 1 | Arbitration Priority Frame DMA | Page 541 |
| PL3 | 0x00000008 | 1 | Arbitration Priority External CPU | Page 541 |
| WT_EN | 0x0000004C | 1 | Weighted-Token Arbitration Scheme Enable | Page 541 |
| WT_TCL | 0x00000050 | 1 | Clock Tokens Refresh Period | Page 542 |
| WT_CL1 | 0x00000054 | 1 | Clock Tokens CPU | Page 542 |
| WT_CL2 | 0x00000058 | 1 | Clock Tokens Frame DMA | Page 542 |
| WT_CL3 | 0x0000005C | 1 | Clock Tokens External CPU | Page 543 |

### 7.18.1.1 SBA:SBA:PL1

**Parent:** SBA:SBA

**Instances:** 1

*Table 723* • **Fields in PL1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PL1 | 3:0 | R/W | Arbitration priority for CPU. Values 0x1 through 0xF, higher value is prioritized over lower value. | 0xE |

### 7.18.1.2 SBA:SBA:PL2

**Parent:** SBA:SBA

**Instances:** 1

*Table 724* • **Fields in PL2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PL2 | 3:0 | R/W | Arbitration priority for Frame DMA. Values 0x1 through 0xF, higher value is prioritized over lower value. | 0xD |

### 7.18.1.3 SBA:SBA:PL3

**Parent:** SBA:SBA

**Instances:** 1

*Table 725* • **Fields in PL3**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PL3 | 3:0 | R/W | Arbitration priority for External CPU. Values 0x1 through 0xF, higher value is prioritized over lower value. | 0xC |

### 7.18.1.4 SBA:SBA:WT_EN

**Parent:** SBA:SBA

**Instances:** 1

When weighted token arbitration is enabled, each master on the shared bus is granted a configurable number of tokens at the start of each refresh period. The length of each refresh period is configurable. In each clock-cycle that a master uses the bus, the token counter for that master decreases. Once all tokens are spent, the master is forced to a low priority. A master with tokens remaining, always takes priority over masters with no tokens remaining.

*Table 726* • **Fields in WT_EN**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WT_EN | 0 | R/W | Set this field to enable weighted-token arbitration scheme. | 0x0 |

### 7.18.1.5 SBA:SBA:WT_TCL

**Parent:** SBA:SBA

**Instances:** 1

*Table 727* • **Fields in WT_TCL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WT_TCL | 15:0 | R/W | Refresh period length for the weighted-token arbitration scheme. | 0xFFFF |

### 7.18.1.6 SBA:SBA:WT_CL1

**Parent:** SBA:SBA

**Instances:** 1

*Table 728* • **Fields in WT_CL1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WT_CL1 | 15:0 | R/W | Number of tokens the CPU is granted at the start of each refresh period for weighted-token arbitration scheme. If configured with a value of zero, the master is considered to have infinite tokens. | 0x0FFF |

### 7.18.1.7 SBA:SBA:WT_CL2

**Parent:** SBA:SBA

**Instances:** 1

*Table 729* • **Fields in WT_CL2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WT_CL2 | 15:0 | R/W | Number of tokens the Frame DMA is granted at the start of each refresh period for weighted-token arbitration scheme. If configured with a value of zero, the master is considered to have infinite tokens. | 0x0FFF |

### 7.18.1.8 SBA:SBA:WT_CL3

**Parent:** SBA:SBA

**Instances:** 1

***Table 730 •*** **Fields in WT_CL3**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| WT_CL3 | 15:0 | R/W | Number of tokens the External CPU is granted at the start of each refresh period for weighted-token arbitration scheme. If configured with a value of zero, the master is considered to have infinite tokens. | 0x0FFF |

## 7.19 GPDMA

***Table 731 •*** **Register Groups in GPDMA**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| CH | 0x00000000 | 8<br>0x00000058 | DMA Channel Controller Configuration | Page 543 |
| INTR | 0x000002C0 | 1 | DMA Interrupt Configuration | Page 555 |
| MISC | 0x00000398 | 1 | Miscellaneous FDMA Registers | Page 561 |

### 7.19.1 GPDMA:CH

**Parent:** GPDMA

**Instances:** 8

***Table 732 •*** **Registers in CH**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SAR | 0x00000000 | 1 | Source Address | Page 544 |
| DAR | 0x00000008 | 1 | Destination Address | Page 544 |
| LLP | 0x00000010 | 1 | Linked List Pointer | Page 545 |
| CTL0 | 0x00000018 | 1 | DMA Transfer Control | Page 545 |
| CTL1 | 0x0000001C | 1 | DMA Transfer Control | Page 548 |
| SSTAT | 0x00000020 | 1 | Source Status | Page 549 |
| DSTAT | 0x00000028 | 1 | Destination Status | Page 549 |

*Table 732 •* **Registers in CH *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| SSTATAR | 0x00000030 | 1 | Source Status Address Location | Page 550 |
| DSTATAR | 0x00000038 | 1 | Destination Status Address Location | Page 550 |
| CFG0 | 0x00000040 | 1 | DMA Transfer Configuration (CFG0) | Page 551 |
| CFG1 | 0x00000044 | 1 | DMA Transfer Configuration (CFG1) | Page 553 |

### 7.19.1.1 GPDMA:CH:SAR

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

*Table 733 •* **Fields in SAR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SAR | 31:0 | R/W | GP (block chaining disabled): Holds the source address aligned to the source transfer width CTL0::SRC_TR_WIDTH of the data to be moved. If the address is not aligned with the source transfer width, H/W auto-aligns. The Current Source Address of DMA transfer is incremented, decremented, or left unchanged on every source transfer throughout the block transfer based on CTL0::SINC. | 0x00000000 |

### 7.19.1.2 GPDMA:CH:DAR

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

*Table 734 •* **Fields in DAR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DAR | 31:0 | R/W | GP (block chaining disabled): Holds the Destination address aligned to the destination transfer width CTL0::DST_TR_WIDTH of the data to be moved. If the address is not aligned with the destination transfer width, H/W auto-aligns. The Current Destination Address of DMA transfer is incremented, decremented, or left unchanged on every source transfer throughout the block transfer based on CTL0::DINC. | 0x00000000 |

### 7.19.1.3 GPDMA:CH:LLP

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

*Table 735 •* **Fields in LLP**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LOC | 30:2 | R/W | Write the 32-bit aligned address of the first DCB in the chain of DCBs. The DMA channel updates this field as it traverses the list of DCBs. The two least significant bits are zeroed out before being used.<br>0 : Disable block chaining (initial read of DCB addressed by LLP before a block transfer)<br>>0: Enable block chaining (initial read of DCB addressed by LLP before a block transfer) | 0x00000000 |

### 7.19.1.4 GPDMA:CH:CTL0

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

*Table 736* • **Fields in CTL0**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LLP_SRC_EN | 28 | R/W | Enable reload of SAR from next DCB in chain. When this field is set and the LLP is non-zero, the SAR will be reloaded from the next DCB upon completion of the current DCB.<br>0: Disable update<br>1: Enable | 0x0 |
| LLP_DST_EN | 27 | R/W | Enable reload of DAR from next DCB in chain. When this field is set and the LLP is non-zero, the DAR will be reloaded from the next DCB upon completion of the current DCB.<br>0: Disable<br>1: Enable | 0x0 |
| SMS | 26:25 | R/W | Source Master Select.<br>INJ/GP: Must be set to 0<br>XTR: Must be set to 1<br>0 = AHB master 1<br>1 = AHB master 2<br>Other: reserved | 0x0 |
| DMS | 24:23 | R/W | Destination Master Select.<br>XTR/GP: Must be set to 0<br>INJ: Must be set to 1<br>0 = AHB master 1<br>1 = AHB master 2<br>Other: Reserved | 0x0 |
| TT_FC | 22:20 | R/W | Transfer Type and Flow Control.<br>GP: Must be set to 0<br>INJ: Must be set to 0 or 1<br>XTR: Must be set to 4<br>0 : Memory to Memory<br>1 : Memory to Peripheral<br>4 : Peripheral to Memory<br>Other: Reserved | 0x3 |

*Table 736 •* **Fields in CTL0** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SRC_MSIZE | 16:14 | R/W | Source Burst Transaction Length.<br>INJ/GP: Number of data items, each with a width of CTL.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface.<br>XTR : Must be <3<br>0 : 1 word<br>1 : 4 words<br>2 : 8 words<br>3: 16 words<br>4: 32 words<br>5: 64 words<br>6: 128 words<br>7: 256 words | 0x1 |
| DEST_MSIZE | 13:11 | R/W | Destination Burst Transaction Length.<br>INJ/GP: Number of data items, each with a width of CTL.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface.<br>XTR : Must be <3<br>0 : 1 word<br>1 : 4 words<br>2 : 8 words<br>3: 16 words<br>4: 32 words<br>5: 64 words<br>6: 128 words<br>7: 256 words | 0x1 |
| SINC | 10:9 | R/W | Source Address Increment.<br>INJ/GP: Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to no change.<br>XTR: Must be set to no change.<br>00 : Increment<br>01 : Decrement<br>1x : No change | 0x0 |

*Table 736 •* **Fields in CTL0** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DINC | 8:7 | R/W | Destination Address Increment.<br>XTR/GP: Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to no change.<br>INJ: Must be set to no change.<br>00 : Increment<br>01 : Decrement<br>1x : No change | 0x0 |
| SRC_TR_WIDTH | 6:4 | R/W | Source Transfer Width.<br>GP: Specifies source address alignment (for example, 32-bit transfer can only be 32-bit aligned).<br>INJ/ XTR: Must be set to 2.<br>0 : 8-bit<br>1 : 16-bit<br>2 : 32-bit<br>Other : Undefined | 0x0 |
| DST_TR_WIDTH | 3:1 | R/W | Destination Transfer Width.<br>GP: Specifies destination address alignment (for example, 32-bit transfer can only be 32-bit aligned).<br>INJ/ XTR: Must be set to 2.<br>0 : 8-bit<br>1 : 16-bit<br>2 : 32-bit<br>Other : Undefined | 0x0 |
| INT_EN | 0 | R/W | Interrupt Enable Bit. If set, then all interrupt-generating sources are enabled.<br>0: Disable<br>1: Enable | 0x1 |

### 7.19.1.5 GPDMA:CH:CTL1

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

If status write-back is enabled, the register is used to update the control register location of the DCB in system memory at the end of the block transfer.

*Table 737 •* **Fields in CTL1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DONE | 12 | R/W | Done bit.<br>Software can poll the DCB CTL.DONE bit to see when a block transfer is complete. The DCB CTL.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.<br>0: Block transfer is not done<br>0: Block transfer is done | 0x0 |
| BLOCK_TS | 11:0 | R/W | Block Transfer Size.<br>INJ/GP : The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer.<br>XTR: Updated with the number of 32-bits words returned.<br>Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral. | 0x002 |

### 7.19.1.6 GPDMA:CH:SSTAT

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

*Table 738 •* **Fields in SSTAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SSTAT | 31:0 | R/W | GP: After each block transfer completes, the source status information can be retrieved from the address to which the contents of the SSTATAR register point. This retrieval is enabled in CFG0.SS_UPD_EN. Once retrieved, the status information is stored in the SSTAT register and written out to the DCB SSTAT register before the start of the next block.<br>INJ/XTR : Must not be used. | 0x00000000 |

### 7.19.1.7 GPDMA:CH:DSTAT

**Parent:** GPDMA:CH

**Instances:** 1

This register is part of the DCB (and is updated on a DCB-by-DCB basis) when block chaining is enabled.

***Table 739 •*** **Fields in DSTAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DSTAT | 31:0 | R/W | After each block transfer completes, the destination status information can be retrieved from the address to which the contents of the DSTATAR register point. This retrieval is enabled in CFG0.DS_UPD_EN. Once retrieved, the status information is stored in the DSTAT register and written out to the DCB DSTAT register before the start of the next block.<br>INJ : Must not be used. | 0x00000000 |

### 7.19.1.8 GPDMA:CH:SSTATAR

**Parent:** GPDMA:CH

**Instances:** 1

***Table 740 •*** **Fields in SSTATAR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SSTATAR | 31:0 | R/W | Specifies the address (if enabled by CFG0.SS_UPD_EN) from where to fetch the source status information, which is registered in the SSTAT register and written out to the DCB SSTAT before the start of the next block. | 0x00000000 |

### 7.19.1.9 GPDMA:CH:DSTATAR

**Parent:** GPDMA:CH

**Instances:** 1

***Table 741 •*** **Fields in DSTATAR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DSTATAR | 31:0 | R/W | Specifies the address (if enabled by CFG0.DS_UPD_EN) from where to fetch the destination status information, which is registered in the DSTAT register and written out to the DCB DSTAT before the start of the next block. | 0x00000000 |

### 7.19.1.10 GPDMA:CH:CFG0

**Parent:** GPDMA:CH

**Instances:** 1

This register contains fields that configure the DMA transfer and remains fixed for all blocks of a multi-block transfer.

***Table 742 •*** **Fields in CFG0**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RELOAD_DST | 31 | R/W | GP: Automatic destination reload. The DAR register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.<br>INJ/XTR : Must be zero.<br>0 : Disable<br>1: Enable | 0x0 |
| RELOAD_SRC | 30 | R/W | GP: Automatic source reload. The SAR register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.<br>INJ/XTR : Must be zero. | 0x0 |
| LOCK_B | 17 | R/W | Bus lock bit. When active, the AHB bus master signal block is asserted for the duration specified in CFG.LOCK_B_L. | 0x0 |
| LOCK_CH | 16 | R/W | Channel lock bit. When the channel is granted control of the master bus interface and if the CFG0.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFG0.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFG0.LOCK_CH_L. | 0x0 |
| LOCK_B_L | 15:14 | R/W | Bus lock level. Indicates the duration over which CFG0.LOCK_B bit applies.<br>0 : Over complete DMA transfer<br>1 : Over complete DMA block transfer<br>Other: Over complete DMA transaction | 0x0 |

*Table 742 •* **Fields in CFG0** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LOCK_CH_L | 13:12 | R/W | Channel lock level. Indicates the duration over which CFG0.LOCK_CH bit applies.<br>0 : Over complete DMA transfer<br>1 : Over complete DMA block transfer<br>Other : Over complete DMA transaction | 0x0 |
| HS_SEL_SRC | 11 | R/W | Source software or hardware handshaking select.<br>INJ/GP : Must be 1<br>XTR: Must be 0<br>0 = Hardware handshaking interface. Software-initiated transaction requests are ignored.<br>1 = Software handshaking interface. Hardware-initiated transaction requests are ignored.<br>If the source peripheral is memory, then this bit is ignored. | 0x1 |
| HS_SEL_DST | 10 | R/W | Destination software or hardware handshaking select.<br>This register selects which of the handshaking interfaces (hardware or software) is active for destination requests on this channel.<br>XTR/GP : Must be 1<br>0 = Hardware handshaking interface. Software-initiated transaction requests are ignored.<br>1 = Software handshaking interface. Hardware-initiated transaction requests are ignored.<br>If the destination peripheral is memory, then this bit is ignored. | 0x1 |
| FIFO_EMPTY | 9 | R/O | Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFG0.CH_SUSP to cleanly disable a channel.<br>1 = Channel FIFO empty<br>0 = Channel FIFO not empty | 0x0 |

*Table 742 •* **Fields in CFG0** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CH_SUSP | 8 | R/W | Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFG0.FIFO_EMPTY to cleanly disable a channel without losing any data.<br>0 = Not suspended<br>1 = Suspend DMA transfer from the source | 0x0 |
| CH_PRIOR | 7:5 | R/W | Channel priority.<br>0 : Lowest priority<br>...<br>7 : Highest priority | 0x0 |

### 7.19.1.11 GPDMA:CH:CFG1

**Parent:** GPDMA:CH

**Instances:** 1

This register contains fields that configure the DMA transfer and remains fixed for all blocks of a multi-block transfer.

*Table 743 •* **Fields in CFG1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DST_PER | 14:11 | R/W | INJ: Destination peripheral handshaking interface. Valid if CFG0.HS_SEL_DST field is 0. Otherwise, this field is ignored.<br>XTR/GP: Not used<br>Must be mapped according the channel number, that is, channel number 0 must be assigned interface 0, and so on. | 0x0 |
| SRC_PER | 10:7 | R/W | XTR: Source peripheral handshaking interface. Valid if CFG0.HS_SEL_SRC field is 0. Otherwise, this field is ignored.<br>INJ/GP: Not used<br>Must be mapped according the channel number, that is, channel number 0 must be assigned interface 0, and so on. | 0x0 |

*Table 743 •* **Fields in CFG1** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SS_UPD_EN | 6 | R/W | Source status update enable.<br>GP: Source status information is fetched only from the location pointed to by the SSTATAR register, stored in the SSTAT register, and written out to the DCB SSTAT if SS_UPD_EN is high.<br>INJ/XTR : Must be zero<br>0: Disable<br>1: Enable | 0x0 |
| DS_UPD_EN | 5 | R/W | Destination status update enable.<br>GP: Destination status information is fetched from the location pointed to by the DSTATAR register, stored in the DSTAT register, and written out to the DCB DSTAT only if DS_UPD_EN is high.<br>INJ : Must be zero<br>XTR : Must be one<br>0: Disable<br>1: Enable | 0x0 |
| RESERVED | 4:2 | R/W | Must be set to its default. | 0x1 |
| FIFOMODE | 1 | R/W | FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.<br>0 = Space/data available for single AHB transfer of the specified transfer width.<br>1 = Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers. The exceptions are at the end of a burst transaction request or at the end of a block transfer. | 0x0 |

*Table 743 •* **Fields in CFG1 *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FCMODE | 0 | R/W | Flow control mode.<br>GP : Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.<br>INJ/XTR : Must be one<br>0 = Source transaction requests are serviced when they occur. Data pre-fetching is enabled.<br>1 = Source transaction requests are not serviced until a destination transaction request occurs. In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled. | 0x0 |

## 7.19.2 GPDMA:INTR

**Parent:** GPDMA

**Instances:** 1

*Table 744 •* **Registers in INTR**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| RAW_TFR | 0x00000000 | 1 | Raw Status for IntTfr Interrupt | Page 556 |
| RAW_BLOCK | 0x00000008 | 1 | Raw Status for IntBlock Interrupt | Page 556 |
| RAW_ERR | 0x00000020 | 1 | Raw Status for IntErr Interrupt | Page 556 |
| STATUS_TFR | 0x00000028 | 1 | Status for IntTfr Interrupt | Page 557 |
| STATUS_BLOCK | 0x00000030 | 1 | Status for IntBlock Interrupt | Page 557 |
| STATUS_ERR | 0x00000048 | 1 | Status for IntErr Interrupt | Page 558 |
| MASK_TFR | 0x00000050 | 1 | Mask for IntTfr Interrupt | Page 558 |
| MASK_BLOCK | 0x00000058 | 1 | Mask for IntBlock Interrupt | Page 558 |
| MASK_ERR | 0x00000070 | 1 | Mask for IntErr Interrupt | Page 559 |
| CLEAR_TFR | 0x00000078 | 1 | Clear for IntTfr Interrupt | Page 560 |
| CLEAR_BLOCK | 0x00000080 | 1 | Clear for IntBlock Interrupt | Page 560 |
| CLEAR_ERR | 0x00000098 | 1 | Clear for IntErr Interrupt | Page 561 |
| STATUSINT | 0x000000A0 | 1 | Status for each interrupt type | Page 561 |

### 7.19.2.1 GPDMA:INTR:RAW_TFR

**Parent:** GPDMA:INTR

**Instances:** 1

This interrupt is generated when the FDMA reaches the end of a DCB chain (done processing a DCB with LLP field = NULL) or when completing a general purpose operation that does not use DCBs.

***Table 745 •*** **Fields in RAW_TFR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RAW_TFR | 7:0 | R/O | Raw interrupt status. Interrupt events are stored in this field before masking. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No interrupt has occurred.<br>1 : Interrupt condition has occurred. | 0x00 |

### 7.19.2.2 GPDMA:INTR:RAW_BLOCK

**Parent:** GPDMA:INTR

**Instances:** 1

BLOCK: This interrupt is generated when the FDMA has processed one DCB or when completing a general purpose operation that does not use DCBs.

***Table 746 •*** **Fields in RAW_BLOCK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RAW_BLOCK | 7:0 | R/O | Raw interrupt status. Interrupt events are stored in this field before masking. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No interrupt has occurred.<br>1 : Interrupt condition has occurred. | 0x00 |

### 7.19.2.3 GPDMA:INTR:RAW_ERR

**Parent:** GPDMA:INTR

**Instances:** 1

ERR: This interrupt is set if the FDMA receives an error-response on the AHB interface (i.e. accessing un-mapped memory space). This condition will not occur unless the FDMA has been misconfigured.

*Table 747 •* **Fields in RAW_ERR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RAW_ERR | 7:0 | R/O | Raw interrupt status. Interrupt events are stored in this field before masking. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No interrupt has occurred.<br>1 : Interrupt condition has occurred. | 0x00 |

### 7.19.2.4 GPDMA:INTR:STATUS_TFR

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 748 •* **Fields in STATUS_TFR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STATUS_TFR | 7:0 | R/O | Interrupt status. Interrupt events are stored in this field after masking. If an interrupt is no masked, it will propagate to this field. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No interrupt has occurred.<br>1 : Interrupt is active. | 0x00 |

### 7.19.2.5 GPDMA:INTR:STATUS_BLOCK

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 749 •* **Fields in STATUS_BLOCK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STATUS_BLOCK | 7:0 | R/O | Interrupt status. Interrupt events are stored in this field after masking. If an interrupt is no masked, it will propagate to this field. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No interrupt has occurred.<br>1 : Interrupt is active. | 0x00 |

### 7.19.2.6 GPDMA:INTR:STATUS_ERR

**Parent:** GPDMA:INTR

**Instances:** 1

***Table 750 •*** **Fields in STATUS_ERR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| STATUS_ERR | 7:0 | R/O | Interrupt status. Interrupt events are stored in this field after masking. If an interrupt is no masked, it will propagate to this field. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No interrupt has occurred.<br>1 : Interrupt is active. | 0x00 |

### 7.19.2.7 GPDMA:INTR:MASK_TFR

**Parent:** GPDMA:INTR

**Instances:** 1

***Table 751 •*** **Fields in MASK_TFR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_MASK_WE_TFR | 15:8 | One-shot | Interrupt mask write enable. In order to write the INT_MASK_TFR field, the corresponding bit in this field must also be set, i.e. to write bit 3 in INT_MASK_TFR bit 3 in this field must also be set at the same time (during the same register write operation).<br>0 : Writing is disabled.<br>1 : Writing is enabled. | 0x00 |
| INT_MASK_TFR | 7:0 | R/W | Interrupt mask. Setting a bit in this field enables the corresponding interrupt to propagate from RAW_TFT to STATUS_TFR and thus activate interrupt. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>In order to write this field the corresponding bit must be set in INT_MASK_WE_TFR.<br>0 : Interrupt is disabled.<br>1 : Interrupt is enabled. | 0x00 |

### 7.19.2.8 GPDMA:INTR:MASK_BLOCK

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 752* • **Fields in MASK_BLOCK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_MASK_WE_BLOCK | 15:8 | One-shot | Interrupt mask write enable. In order to write the INT_MASK_BLOCK field, the corresponding bit in this field must also be set, i.e. to write bit 3 in INT_MASK_BLOCK bit 3 in this field must also be set at the same time (during the same register write operation).<br>0 : Writing is disabled.<br>1 : Writing is enabled. | 0x00 |
| INT_MASK_BLOCK | 7:0 | R/W | Interrupt mask. Setting a bit in this field enables the corresponding interrupt to propagate from RAW_BLOCK to STATUS_BLOCK and thus activate interrupt. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>In order to write this field the corresponding bit must be set in INT_MASK_WE_BLOCK.<br>0 : Interrupt is disabled.<br>1 : Interrupt is enabled. | 0x00 |

### 7.19.2.9 GPDMA:INTR:MASK_ERR

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 753* • **Fields in MASK_ERR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_MASK_WE_ERR | 15:8 | One-shot | Interrupt mask write enable. In order to write the INT_MASK_ERR field, the corresponding bit in this field must also be set, i.e. to write bit 3 in INT_MASK_ERR bit 3 in this field must also be set at the same time (during the same register write operation).<br>0 : Writing is disabled.<br>1 : Writing is enabled. | 0x00 |

*Table 753 •* **Fields in MASK_ERR *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| INT_MASK_ERR | 7:0 | R/W | Interrupt mask. Setting a bit in this field enables the corresponding interrupt to propagate from RAW_ERR to STATUS_ERR and thus activate interrupt. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>In order to write this field the corresponding bit must be set in INT_MASK_WE_ERR.<br>0 : Interrupt is disabled.<br>1 : Interrupt is enabled. | 0x00 |

### 7.19.2.10 GPDMA:INTR:CLEAR_TFR

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 754 •* **Fields in CLEAR_TFR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLEAR_TFR | 7:0 | One-shot | Interrupt clear. Setting this field clears interrupt indications from the RAW_TFR and STATUS_TFR registers. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No effect.<br>1 : Clear interrupt indication. | 0x00 |

### 7.19.2.11 GPDMA:INTR:CLEAR_BLOCK

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 755 •* **Fields in CLEAR_BLOCK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLEAR_BLOCK | 7:0 | One-shot | Interrupt clear. Setting this field clears interrupt indications from the RAW_BLOCK and STATUS_BLOCK registers. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No effect.<br>1 : Clear interrupt indication. | 0x00 |

### 7.19.2.12 GPDMA:INTR:CLEAR_ERR

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 756* • **Fields in CLEAR_ERR**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CLEAR_ERR | 7:0 | One-shot | Interrupt clear. Setting this field clears interrupt indications from the RAW_ERR and STATUS_ERR registers. Each bit in this field corresponds to a channel in the FDMA, i.e. bit 0 correspond to channel 0, and so on.<br>0 : No effect.<br>1 : Clear interrupt indication. | 0x00 |

### 7.19.2.13 GPDMA:INTR:STATUSINT

**Parent:** GPDMA:INTR

**Instances:** 1

*Table 757* • **Fields in STATUSINT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ERR | 4 | R/O | This field is set if any of the STATUS_ERR.STATUS_ERR interrupts are active.<br>0 : No ERR interrupts are active.<br>1 : At least one ERR interrupt is active. | 0x0 |
| BLOCK | 1 | R/O | This field is set if any of the STATUS_BLOCK.STATUS_BLOCK interrupts are active.<br>0 : No BLOCK interrupts are active.<br>1 : At least one BLOCK interrupt is active. | 0x0 |
| TFR | 0 | R/O | This field is set if any of the STATUS_TFR.STATUS_TFR interrupts are active.<br>0 : No TFR interrupts are active.<br>1 : At least one TFR interrupt is active. | 0x0 |

## 7.19.3 GPDMA:MISC

**Parent:** GPDMA

**Instances:** 1

*Table 758* • **Registers in MISC**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| DMA_CFG_REG | 0x00000000 | 1 | DMA Enable | Page 562 |
| CH_EN_REG | 0x00000008 | 1 | DMA Channel Enable | Page 562 |
| DMA_COMP_VERSION | 0x00000064 | 1 | DMA Version | Page 562 |

### 7.19.3.1 GPDMA:MISC:DMA_CFG_REG

**Parent:** GPDMA:MISC

**Instances:** 1

*Table 759* • **Fields in DMA_CFG_REG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DMA_EN | 0 | R/W | DMA enable bit<br>0: Disable<br>1: Enable | 0x0 |

### 7.19.3.2 GPDMA:MISC:CH_EN_REG

**Parent:** GPDMA:MISC

**Instances:** 1

*Table 760* • **Fields in CH_EN_REG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CH_EN_WE | 15:8 | One-shot | Channel enable write enable | 0x00 |
| CH_EN | 7:0 | R/W | Enables or disables the channel. Setting this bit enables a channel; clearing this bit disables the channel.<br>The bit is automatically cleared by hardware to disable the channel after the last DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.<br>0: Disable the channel<br>1: Enable the channel | 0x00 |

### 7.19.3.3 GPDMA:MISC:DMA_COMP_VERSION

**Parent:** GPDMA:MISC

**Instances:** 1

*Table 761* • **Fields in DMA_COMP_VERSION**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DMA_COMP_VERSION | 31:0 | R/O | Version of the component. | 0x3231342A |

# 7.20 PHY

*Table 762* • **Register Groups in PHY**

| Register Group Name | Offset within Target | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_STD | 0x00000000 | 1 | IEEE Standard and Main Registers | Page 563 |
| PHY_EXT1 | 0x00000000 | 1 | Extended Page 1 Registers | Page 590 |
| PHY_EXT2 | 0x00000000 | 1 | Extended Page 2 Registers | Page 596 |
| PHY_GP | 0x00000000 | 1 | General Purpose Registers | Page 598 |
| PHY_EEE | 0x00000000 | 1 | Clause 45 Registers to Support Energy Efficient | Page 601 |

## 7.20.1 PHY:PHY_STD

**Parent:** PHY

**Instances:** 1

The following section lists the standard register set for the PHY.

*Table 763* • **Registers in PHY_STD**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_CTRL | 0x00000000 | 1 | Control (Address 0) | Page 565 |
| PHY_STAT | 0x00000001 | 1 | Status (Address 1) | Page 566 |
| PHY_IDF1 | 0x00000002 | 1 | PHY Identifier Number 1 (Address 2) | Page 567 |
| PHY_IDF2 | 0x00000003 | 1 | PHY Identifier Number 2 (Address 3) | Page 567 |
| PHY_AUTONEG_ADVERTISMENT | 0x00000004 | 1 | Auto-Negotiation Advertisement (Address 4) | Page 567 |
| PHY_AUTONEG_LP_ABILITY | 0x00000005 | 1 | Auto-Negotiation Link Partner Base Page Ability (Address 5) | Page 568 |
| PHY_AUTONEG_EXP | 0x00000006 | 1 | Auto-Negotiation Expansion (Address 6) | Page 569 |
| PHY_AUTONEG_NEXTPAGE_TX | 0x00000007 | 1 | Auto-Negotiation Next-Page Transmit (Address 7) | Page 569 |

*Table 763 •* **Registers in PHY_STD *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_AUTONEG_LP_NEXTPAGE_RX | 0x00000008 | 1 | Auto-Negotiation Next-Page Receive (Address 8) | Page 569 |
| PHY_CTRL_1000BT | 0x00000009 | 1 | 1000BASE-T Control (Address 9) | Page 570 |
| PHY_STAT_1000BT | 0x0000000A | 1 | 1000BASE-T Status (Address 10) | Page 571 |
| MMD_ACCESS_CFG | 0x0000000D | 1 | MMD Access Control Register (Address 13) | Page 572 |
| MMD_ADDR_DATA | 0x0000000E | 1 | MMD Address or Data Register (Address 14) | Page 572 |
| PHY_STAT_1000BT_EXT1 | 0x0000000F | 1 | 1000BASE-T Status Extension Number 1 (Address 15) | Page 573 |
| PHY_STAT_100BTX | 0x00000010 | 1 | 100BASE-TX Status (Address 16) | Page 573 |
| PHY_STAT_1000BT_EXT2 | 0x00000011 | 1 | 1000BASE-T Status Extension Number 2 (Address 17) | Page 574 |
| PHY_BYPASS_CTRL | 0x00000012 | 1 | Bypass Control (Address 18) | Page 575 |
| PHY_ERROR_CNT1 | 0x00000013 | 1 | Error Counter Number 1 (Address 19) | Page 576 |
| PHY_ERROR_CNT2 | 0x00000014 | 1 | Error Counter Number 2 (Address 20) | Page 576 |
| PHY_ERROR_CNT3 | 0x00000015 | 1 | Error Counter Number 3 (Address 21) | Page 577 |
| PHY_CTRL_STAT_EXT | 0x00000016 | 1 | Extended Control and Status (Address 22) | Page 577 |
| PHY_CTRL_EXT1 | 0x00000017 | 1 | Extended Control Number 1 (Address 23) | Page 579 |
| PHY_CTRL_EXT2 | 0x00000018 | 1 | Extended Control Number 2 (Address 24) | Page 579 |
| PHY_INT_MASK | 0x00000019 | 1 | Interrupt Mask (Address 25) | Page 581 |
| PHY_INT_STAT | 0x0000001A | 1 | Interrupt Status (Address 26) | Page 583 |
| PHY_AUX_CTRL_STAT | 0x0000001C | 1 | Auxiliary Control and Status (Address 28) | Page 585 |
| PHY_LED_MODE_SEL | 0x0000001D | 1 | LED Mode Select (Address 29) | Page 588 |
| PHY_LED_BEHAVIOR_CTRL | 0x0000001E | 1 | LED Behavior Control (Address 30) | Page 588 |
| PHY_MEMORY_PAGE_ACCESS | 0x0000001F | 1 | Memory Page Access (Address 31) | Page 590 |

### 7.20.1.1 PHY:PHY_STD:PHY_CTRL

**Parent:** PHY:PHY_STD

**Instances:** 1

***Table 764 •*** **Fields in PHY_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SOFTWARE_RESET_ENA | 15 | R/W | Initiate software reset. This field is cleared as part of this operation. After enabling this field, you must wait at least 4 us before PHY registers can be accessed again. | 0x0 |
| LOOPBACK_ENA | 14 | R/W | Enable loopback mode. The loopback mechanism works at the current speed. If the link is down (see PHY_STAT.LINK_STATUS), SPEED_SEL_LSB_CFG and SPEED_SEL_MSB_CFG determine the operating speed of the loopback. | 0x0 |
| SPEED_SEL_LSB_CFG | 13 | R/W | Least significant bit of the speed selection, along with SPEED_SEL_MSB_CFG, this field determines the speed when auto-negotiation is disabled (See AUTONEG_ENA).<br>00: 10 Mbps<br>01: 100 Mbps<br>10: 1000 Mbps<br>11: Reserved | 0x0 |
| AUTONEG_ENA | 12 | R/W | Enable auto-negotiation. When cleared, the speed and duplex-mode are determined by SPEED_SEL_LSB_CFG, SPEED_SEL_MSB_CFG, and DUPLEX_MODE_CFG. | 0x1 |
| POWER_DOWN_ENA | 11 | R/W | Enable power-down mode. This disables PHY operation until this bit is cleared or the PHY is reset. | 0x0 |
| ISOLATE_ENA | 10 | R/W | Isolate the PHY from the integrated MAC. | 0x0 |
| AUTONEG_RESTART_ENA | 9 | R/W | Restart an auto-negotiation cycle; the PHY clears this field when auto-negotiation is restarted. | 0x0 |
| DUPLEX_MODE_CFG | 8 | R/W | Configure duplex mode when auto-negotiation is disabled (see AUTONEG_ENA).<br>0: Half-duplex<br>1: Full-duplex | 0x0 |

*Table 764 •* **Fields in PHY_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| COLLISION_TEST_ENA | 7 | R/W | Enable collision indication test-mode, when enabled the PHY indicate collision when the MAC transmits data to the PHY. | 0x0 |
| SPEED_SEL_MSB_CFG | 6 | R/W | See SPEED_SEL_LSB_CFG. | 0x1 |

### 7.20.1.2 PHY:PHY_STD:PHY_STAT

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 765 •* **Fields in PHY_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MODE_100BT4 | 15 | R/O | The PHY is not 100BASE-T4 capable. | 0x0 |
| MODE_100BX_FDX | 14 | R/O | The PHY is 100BASE-X FDX capable. | 0x1 |
| MODE_100BX_HDX | 13 | R/O | The PHY is 100BASE-X HDX capable. | 0x1 |
| MODE_10BT_FDX | 12 | R/O | The PHY is 10BASE-T FDX capable. | 0x1 |
| MODE_10BT_HDX | 11 | R/O | The PHY is 10BASE-T HDX capable. | 0x1 |
| MODE_100BT2_FDX | 10 | R/O | The PHY is not 100BASE-T2 FDX capable. | 0x0 |
| MODE_100BT2_HDX | 9 | R/O | The PHY is not 100BASE-T2 HDX capable. | 0x0 |
| EXT_STATUS | 8 | R/O | Extended status information are available; see the PHY_STAT_EXT register. | 0x1 |
| PREAMBLE_SUPPRESS | 6 | R/O | The PHY accepts management frames with preamble suppressed. | 0x1 |
| AUTONEG_COMPLETE | 5 | R/O | This field is set when auto-negotiation is completed and cleared during active auto-negotiation cycles. | 0x0 |
| REMOTE_FAULT | 4 | R/O | This field is set when the PHY detects a remote fault condition and cleared on register read. | 0x0 |
| AUTONEG_ABILITY | 3 | R/O | The PHY is capable of auto-negotiation. | 0x1 |
| LINK_STAT | 2 | R/O | This field is cleared when the link is down. It is set when the link is up and a previous link-down indication was read from the register. | 0x0 |

*Table 765 •* **Fields in PHY_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| JABBER_DETECT | 1 | R/O | This field is set when the PHY detects a jabber condition and cleared on register read. | 0x0 |
| EXT_CAPABILITY | 0 | R/O | The PHY provides an extended set of capabilities. | 0x1 |

### 7.20.1.3 PHY:PHY_STD:PHY_IDF1

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 766 •* **Fields in PHY_IDF1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OUI_MS | 15:0 | R/O | Vitesse's organizationally unique identifier bits 3 through 18. | 0x0007 |

### 7.20.1.4 PHY:PHY_STD:PHY_IDF2

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 767 •* **Fields in PHY_IDF2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| OUI_LS | 15:10 | R/O | Vitesse's organizationally unique identifier bits 19 through 24. | 0x01 |
| MODEL_NUMBER | 9:4 | R/O | The device model number. | 0x2D |
| REVISION_NUMBER | 3:0 | R/O | The device revision number. | 0x0 |

### 7.20.1.5 PHY:PHY_STD:PHY_AUTONEG_ADVERTISMENT

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 768 •* **Fields in PHY_AUTONEG_ADVERTISMENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NEXT_PAGE_ENA | 15 | R/W | Advertises desire to engage in next-page exchange. When this field is set, next-page control is returned to the user for additional next-pages following the 1000BASE-T next-page exchange. | 0x0 |
| REMOTE_FAULT_CFG | 13 | R/W | Transmit Remote Fault. | 0x0 |
| ASYM_PAUSE_CFG | 11 | R/W | Advertise asymmetric pause capability. | 0x0 |

*Table 768 •* **Fields in PHY_AUTONEG_ADVERTISMENT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SYM_PAUSE_CFG | 10 | R/W | Advertise symmetric pause capability. | 0x0 |
| ADV_100BT4_CFG | 9 | R/W | Advertise 100BASE-T4 capability. | 0x0 |
| ADV_100BX_FDX_CFG | 8 | R/W | Advertise 100BASE-X FDX capability. | 0x1 |
| ADV_100BX_HDX_CFG | 7 | R/W | Advertise 100BASE-X HDX capability. | 0x1 |
| ADV_10BT_FDX_CFG | 6 | R/W | Advertise 10BASE-T FDX capability. | 0x1 |
| ADV_10BT_HDX_CFG | 5 | R/W | Advertise 10BASE-T HDX capability. | 0x1 |
| SELECTOR_FIELD_CFG | 4:0 | R/W | Select types of message send by auto-negotiation. | 0x01 |

### 7.20.1.6 PHY:PHY_STD:PHY_AUTONEG_LP_ABILITY

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 769 •* **Fields in PHY_AUTONEG_LP_ABILITY**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LP_NEXT_PAGE | 15 | R/O | Link partner advertises desire to engage in next-page exchange. | 0x0 |
| LP_ACKNOWLEDGE | 14 | R/O | Link partner advertises that link code word was successfully received. | 0x0 |
| LP_REMOTE_FAULT | 13 | R/O | Link partner advertises remote fault. | 0x0 |
| LP_ASYM_PAUSE | 11 | R/O | Link partner advertises asymmetric pause capability. | 0x0 |
| LP_SYM_PAUSE | 10 | R/O | Link partner advertises symmetric pause capability. | 0x0 |
| LP_100BT4 | 9 | R/O | Link partner advertises 100BASE-T4 capability. | 0x0 |
| LP_100BX_FDX | 8 | R/O | Link partner advertises 100BASE-X FDX capability. | 0x0 |
| LP_100BX_HDX | 7 | R/O | Link partner advertises 100BASE-X HDX capability. | 0x0 |
| LP_10BT_FDX | 6 | R/O | Link partner advertises 10BASE-T FDX capability. | 0x0 |
| LP_10BT_HDX | 5 | R/O | Link partner advertises 10BASE-T HDX capability. | 0x0 |
| LP_SELECTOR_FIELD | 4:0 | R/O | Link partner advertises select type of message send by auto-negotiation. | 0x00 |

### 7.20.1.7 PHY:PHY_STD:PHY_AUTONEG_EXP

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 770* • **Fields in PHY_AUTONEG_EXP**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PARALLEL_DET_FAULT | 4 | R/O | This field is set when the PHY detects a Receive Link Integrity Test Failure condition and cleared on register read. | 0x0 |
| LP_NEXT_PAGE_ABLE | 3 | R/O | Set if link partner is next-page capable. | 0x0 |
| NEXT_PAGE_ABLE | 2 | R/O | The PHY is next-page capable. | 0x1 |
| NEXT_PAGE_RECEIVED | 1 | R/O | This field is set when the PHY receives a valid next-page and cleared on register read. | 0x0 |
| LP_AUTONEG_ABLE | 0 | R/O | Set if link partner is auto-negotiation capable. | 0x0 |

### 7.20.1.8 PHY:PHY_STD:PHY_AUTONEG_NEXTPAGE_TX

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 771* • **Fields in PHY_AUTONEG_NEXTPAGE_TX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NEXT_PAGE_CFG | 15 | R/W | Set to indicate that more pages will follow; clear if current page is the last. | 0x0 |
| MESSAGE_PAGE_CFG | 13 | R/W | Set to indicate that this is a message page; clear if the current page consists of unformatted code. | 0x1 |
| ACKNOWLEDGE2_CFG | 12 | R/W | Set to indicate ability to comply with the request of the last received page. | 0x0 |
| TOGGLE | 11 | R/O | Alternates between 0 and 1 for each transmitted page. | 0x0 |
| MESSAGE_FIELD_CFG | 10:0 | R/W | Contains page information - either message or unformatted code. MESSAGE_PAGE_CFG must indicate if this page contains either a message or unformatted code. | 0x001 |

### 7.20.1.9 PHY:PHY_STD:PHY_AUTONEG_LP_NEXTPAGE_RX

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 772 •* **Fields in PHY_AUTONEG_LP_NEXTPAGE_RX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LP_NEXT_PAGE_RX | 15 | R/O | Set by link partner to indicate that more pages follow. When cleared, this is the last of the next-pages. | 0x0 |
| LP_ACKNOWLEDGE_RX | 14 | R/O | Set by link partner to acknowledge the reception of last message. | 0x0 |
| LP_MESSAGE_PAGE | 13 | R/O | Set by Link partner if this page contains a message. When cleared this page contains unformatted code. | 0x0 |
| LP_ACKNOWLEDGE2 | 12 | R/O | Set by link partner to indicate that it is able to act on transmitted information. | 0x0 |
| LP_TOGGLE | 11 | R/O | Will alternate between 0 and 1 for each received page. Used to check for errors. | 0x0 |
| LP_MESSAGE_FIELD | 10:0 | R/O | Contains page information, MESSAGE_PAGE indicates if this page contains either a message or unformatted code. | 0x000 |

### 7.20.1.10 PHY:PHY_STD:PHY_CTRL_1000BT

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 773 •* **Fields in PHY_CTRL_1000BT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_TEST_MODE_CFG | 15:13 | R/W | Configure 1000BASE-T test modes; this field is only valid in 1000BASE-T mode. Other encodings are reserved and must not be selected.<br>0: Normal operation<br>1: Transmit waveform test.<br>2: Transmit jitter test in master mode.<br>3: Transmit jitter test in slave mode.<br>4: Transmit distortion test. | 0x0 |
| MS_MANUAL_CFG_ENA | 12 | R/W | Enable manual configuration of master/slave value. | 0x0 |

*Table 773 •* **Fields in PHY_CTRL_1000BT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MS_MANUAL_CFG | 11 | R/W | Configure if the PHY should configure itself as either master or slave during master/slave negotiations. This field is only valid when MS_MANUAL_CFG_ENA is set.<br>0: Configure as slave.<br>1: Configure as master. | 0x0 |
| PORT_TYPE_CFG | 10 | R/W | Set to indicate multi-port device, clear to indicate single-port device. | 0x1 |
| ADV_1000BT_FDX_CFG | 9 | R/W | Set to advertise 1000BASE-T FDX capability. | 0x1 |
| ADV_1000BT_HDX_CFG | 8 | R/W | Set to advertise 1000BASE-T HDX capability. | 0x1 |

### 7.20.1.11 PHY:PHY_STD:PHY_STAT_1000BT

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 774 •* **Fields in PHY_STAT_1000BT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MS_CFG_FAULT | 15 | R/O | This field is set when the PHY detects a master/slave configuration fault condition and cleared on register read. | 0x0 |
| MS_CFG_RESOLUTION | 14 | R/O | This field indicates the result of a master/slave Negotiation.<br>0: Local PHY is resolved to slave.<br>1: Local PHY is resolved to master. | 0x1 |
| LOCAL_RECEIVER_STAT | 13 | R/O | The status of the local receiver (loc_rcvr_status as defined in IEEE Std. 802.3).<br>0: Local receiver status is NOT_OK.<br>1: Local receiver status is OK. | 0x0 |
| REMOTE_RECEIVER_STAT | 12 | R/O | The status of the remote receiver (rem_rcvr_status as defined in IEEE Std. 802.3).<br>0: Remote receiver status is NOT_OK.<br>1: Remote receiver status is OK. | 0x0 |
| LP_1000BT_FDX | 11 | R/O | Set if link partner advertises 1000BASE-T FDX capability. | 0x0 |
| LP_1000BT_HDX | 10 | R/O | Set if link partner advertises 1000BASE-T HDX capability. | 0x0 |

*Table 774 •* **Fields in PHY_STAT_1000BT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| IDLE_ERR_CNT | 7:0 | R/O | Counts each occurrence of rxerror_status = Error (rx_error_status as defined in IEEE Std. 802.3. This field is cleared on read and saturates at all-ones. | 0x00 |

### 7.20.1.12 PHY:PHY_STD:MMD_ACCESS_CFG

**Parent:** PHY:PHY_STD

**Instances:** 1

The bits in this register of the main register space are a window to the EEE registers as defined in IEEE 802.3az Clause 45.

*Table 775 •* **Fields in MMD_ACCESS_CFG**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MMD_FUNCTION | 15:14 | R/W | Function.<br>0: Address<br>1: Data, no post increment<br>2: Data, post increment for read and write<br>3: Data, post increment for write only | 0x0 |
| MMD_DVAD | 4:0 | R/W | Device address as defined in IEEE 802.3az table 45-1. | 0x00 |

### 7.20.1.13 PHY:PHY_STD:MMD_ADDR_DATA

**Parent:** PHY:PHY_STD

**Instances:** 1

The bits in this register of the main register space are a window to the EEE registers as defined in IEEE 802.3az Clause 45.

*Table 776 •* **Fields in MMD_ADDR_DATA**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MMD_ADDR_DATA | 15:0 | R/W | If MMD_ACCESS_CFG.MMD_FUNCTION is 0, MMD_ADDR_DATA specifies the address of register of the device that is specified by MMD_ACCESS_CFG.MMD_DVAD. Otherwise, MMD_ADDR_DATA specifies the data to be written to or read from the register. | 0x0000 |

### 7.20.1.14 PHY:PHY_STD:PHY_STAT_1000BT_EXT1

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 777* • **Fields in PHY_STAT_1000BT_EXT1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MODE_1000BX_FDX | 15 | R/O | The PHY is not 1000BASE-X FDX capable. | 0x0 |
| MODE_1000BX_HDX | 14 | R/O | The PHY is not 1000BASE-X HDX capable. | 0x0 |
| MODE_1000BT_FDX | 13 | R/O | The PHY is 1000BASE-T FDX capable. | 0x1 |
| MODE_1000BT_HDX | 12 | R/O | The PHY is 1000BASE-T HDX capable. | 0x1 |

### 7.20.1.15 PHY:PHY_STD:PHY_STAT_100BTX

**Parent:** PHY:PHY_STD

**Instances:** 1

These fields are only valid in 100BASE-T mode.

*Table 778* • **Fields in PHY_STAT_100BTX**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DESCRAM_LOCKED | 15 | R/O | This field is set when the 100BASE-TX descrambler is in lock and cleared when it is out of lock. | 0x0 |
| DESCRAM_ERR | 14 | R/O | This field is set when the PHY detects a descrambler error condition and cleared on register read. | 0x0 |
| LINK_DISCONNECT | 13 | R/O | This field is set when the PHY detects a 100BASE-TX link disconnect condition and cleared on register read. | 0x0 |
| LINK_STAT_100 | 12 | R/O | This field is set when the 100BASE-TX link status is active and cleared when inactive. | 0x0 |
| RECEIVE_ERR | 11 | R/O | This field is set when the PHY detects a receive error condition and cleared on register read. | 0x0 |
| TRANSMIT_ERR | 10 | R/O | This field is set when the PHY detects a transmit error condition and cleared on register read. | 0x0 |

*Table 778 •* **Fields in PHY_STAT_100BTX *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SSD_ERR | 9 | R/O | This field is set when the PHY detects a Start-of-Stream Delimiter Error condition and cleared on register read. | 0x0 |
| ESD_ERR | 8 | R/O | This field is set when the PHY detects an End-of-Stream Delimiter Error condition and cleared on register read. | 0x0 |

### 7.20.1.16 PHY:PHY_STD:PHY_STAT_1000BT_EXT2

**Parent:** PHY:PHY_STD

**Instances:** 1

These fields are only valid in 1000BASE-T mode.

*Table 779 •* **Fields in PHY_STAT_1000BT_EXT2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| DESCRAM_LOCKED_1000 | 15 | R/O | This field is set when the 1000BASE-T descrambler is in lock and cleared when it is out of lock. | 0x0 |
| DESCRAM_ERR_1000 | 14 | R/O | This field is set when the PHY detects a Descrambler Error condition and cleared on register read. | 0x0 |
| LINK_DISCONNECT_1000 | 13 | R/O | This field is set when the PHY detects a 1000BASE-T link disconnect condition and cleared on register read. | 0x0 |
| LINK_STAT_1000 | 12 | R/O | This field is set when the 1000BASE-T link status is active and cleared when inactive. | 0x0 |
| RECEIVE_ERR_1000 | 11 | R/O | This field is set when the PHY detects a Receive Error condition and cleared on register read. | 0x0 |
| TRANSMIT_ERR_1000 | 10 | R/O | This field is set when the PHY detects a Transmit Error condition and cleared on register read. | 0x0 |
| SSD_ERR_1000 | 9 | R/O | This field is set when the PHY detects a Start-of-Stream Delimiter Error condition and cleared on register read. | 0x0 |
| ESD_ERR_1000 | 8 | R/O | This field is set when the PHY detects an End-of-Stream Delimiter Error condition and cleared on register read. | 0x0 |

*Table 779 •* **Fields in PHY_STAT_1000BT_EXT2 *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| CARRIER_EXT_ERR_1000 | 7 | R/O | This field is set when the PHY detects a 1000BASE-T Carrier Extension Error condition and cleared on register read. | 0x0 |
| BCM5400_ERR_1000 | 6 | R/O | This field is set when the PHY detects a non-compliant BCM5400 condition. This field is only valid when the 1000BASE-T descrambler is in locked state (see DESCRAM_LOCKED_1000). | 0x0 |
| MDI_CROSSOVER_ERR | 5 | R/O | This field is set when the PHY detects an MDI crossover error condition. | 0x0 |

### 7.20.1.17 PHY:PHY_STD:PHY_BYPASS_CTRL

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 780 •* **Fields in PHY_BYPASS_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_DIS | 15 | R/W | Disable the PHY transmitter. When set, the analog blocks are powered down and zeros are send to the DAC. | 0x0 |
| ENC_DEC_4B5B | 14 | R/W | If set, bypass the 4B5B encoder/decoder. | 0x0 |
| SCRAMBLER | 13 | R/W | If set, bypass the scrambler. | 0x0 |
| DESCRAMBLER | 12 | R/W | If set, bypass the descrambler. | 0x0 |
| PCS_RX | 11 | R/W | If set, bypass the PCS receiver. | 0x0 |
| PCS_TX | 10 | R/W | If set, bypass the PCS transmit. | 0x0 |
| LFI_TIMER | 9 | R/W | If set, bypass the link fail inhibit (LFI) timer. | 0x0 |
| FORCED_SPEED_AUTO_MDIX_DIS | 7 | R/W | Bit for disabling HP AutoMDIX in forced 10/100 speeds, even though auto-negotiation is disabled.<br>0: The HP Auto-MDIX function is enabled.<br>1: Default value. The HP Auto-MDIX function is disabled. Use the default value when in auto-negotiation mode. | 0x1 |

*Table 780 •* **Fields in PHY_BYPASS_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAIR_SWAP_DIS | 5 | R/W | Disable automatic pair swap correction. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| POL_INV_DIS | 4 | R/W | Disable automatic polarity inversion correction. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| PARALLEL_DET_DIS | 3 | R/W | When cleared, the PHY ignores its advertised abilities when performing parallel detect. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x1 |
| PULSE_SHAPING_DIS | 2 | R/W | If set, disable the pulse shaping filter. | 0x0 |
| AUTO_NP_EXCHANGE_DIS | 1 | R/W | Disable automatic exchange of 1000BASE-T next pages. If this feature is disabled, you have the responsibility of sending next pages, determining capabilities, and configuration of the PHY after successful exchange of pages. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |

### 7.20.1.18 PHY:PHY_STD:PHY_ERROR_CNT1

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 781 •* **Fields in PHY_ERROR_CNT1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_ERR_CNT | 7:0 | R/O | Counter containing the number of packets received with errors for 100/1000BASE-TX. The counter saturates at 255 and it is cleared when read. | 0x00 |

### 7.20.1.19 PHY:PHY_STD:PHY_ERROR_CNT2

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 782* • **Fields in PHY_ERROR_CNT2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FALSE_CARRIER_CNT | 7:0 | R/O | Counter containing the number of false carrier incidents for 100/1000BASE-TX. The counter saturates at 255 and it is cleared when read. | 0x00 |

### 7.20.1.20 PHY:PHY_STD:PHY_ERROR_CNT3

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 783* • **Fields in PHY_ERROR_CNT3**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_DIS_CNT | 7:0 | R/O | Counter containing the number of copper media link disconnects. The counter saturates at 255 and it is cleared when read. | 0x00 |

### 7.20.1.21 PHY:PHY_STD:PHY_CTRL_STAT_EXT

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 784* • **Fields in PHY_CTRL_STAT_EXT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_10BT_FORCE_ENA | 15 | R/W | When this field is set, the PHY link integrity state machine is bypassed, and the PHY is forced into link pass status. This is a sticky field; see PHY_CTRL_EXT.STICKY_RESET_ENA. | 0x0 |
| JABBER_DETECT_DIS | 14 | R/W | Disable jabber detect function. When this is disabled, the PHY allows transmission requests to be arbitrarily long without shutting down the transmitter. When cleared, the PHY shuts down the transmitter after the specified time limit specified by IEEE. This is a sticky field; see PHY_CTRL_EXT.STICKY_RESET_ENA. | 0x0 |

*Table 784 •* **Fields in PHY_CTRL_STAT_EXT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ECHO_10BT_DIS | 13 | R/W | When this field is set, the state of the TX_EN pin does not echo onto the CRS pin, which effectively disables CRS from being asserted in half-duplex operation. When cleared, the TX_EN pin is echoed onto the CRS pin. This applies only in 10BASE-T mode. This is a sticky field; see PHY_CTRL_EXT.STICKY_RESET_ENA. | 0x1 |
| SQE_10BT_DIS | 12 | R/W | Disable SQE (Signal Quality Error) pulses on the MAC interface. This applies only in 10BASE-T mode. This is a sticky field; see PHY_CTRL_EXT.STICKY_RESET_ENA. | 0x1 |
| SQUELCH_10BT_CFG | 11:10 | R/W | Configure squelch control (this only applies in the 10BASE-T mode). This is a sticky field; see PHY_CTRL_EXT.STICKY_RESET_ENA.<br>0: The PHY uses the squelch threshold levels prescribed by the IEEE 10BASE-T specification.<br>1: In this mode, the squelch levels are decreased, which may improve the bit error rate performance on long loops<br>2: In this mode, the squelch levels are increased, which may improve the bit error rate in high-noise environments<br>3: Reserved. | 0x0 |
| STICKY_RESET_ENA | 9 | R/W | When set, all fields described as sticky retain their value during software reset. When cleared, all fields marked as sticky are reset to their default values during software reset.<br>This does not affect hardware resets. This is a super-sticky field, which means that it always retain its value during software reset. | 0x1 |
| EOF_ERR | 8 | R/O | When set, this field indicates that a defective EOF (End Of Frame) sequence was received since the last time this field was read. This field is cleared on read. | 0x0 |

*Table 784 •* **Fields in PHY_CTRL_STAT_EXT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_10BT_DISCONNECT | 7 | R/O | When set, this field indicates that the carrier integrity monitor has broken the 10BASE-T connection since the last read of this bit. This field is cleared on read. | 0x0 |
| LINK_10BT_STAT | 6 | R/O | This field is set when a 10BASE-T link is active. Cleared when inactive. | 0x0 |
| BROADCAST_WRITE_ENA | 0 | R/W | Enable any MII write operation (regardless of destination PHY) to be interpreted as a write to this PHY. This only applies to writes; read-operations are still interpreted with correct address. This is particularly useful when similar settings should be propagated to multiple PHYs. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |

### 7.20.1.22 PHY:PHY_STD:PHY_CTRL_EXT1

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 785 •* **Fields in PHY_CTRL_EXT1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 15:4 | R/W | Must be set to its default. | 0x000 |
| FAR_END_LOOPBACK_ENA | 3 | R/W | Enable far end loopback in this PHY. In this mode all incoming traffic on the media interface is retransmitted back to the link partner. In addition, the incoming data also appears on the internal Rx interface to the MAC. Any data send to the PHY from the internal MAC is ignored when this mode is active. | 0x0 |

### 7.20.1.23 PHY:PHY_STD:PHY_CTRL_EXT2

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 786 •* **Fields in PHY_CTRL_EXT2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EDGE_RATE_CFG | 15:13 | R/W | Control the transmit DAC slew rate in 100BASE-TX mode only. The difference between each setting is approximately 200ps to 300ps, with the +3 setting resulting in the slowest edge rate, and the -4 setting resulting in the fastest edge rate. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>011: +5 Edge rate (slowest).<br>010: +4 Edge rate.<br>001: +3 Edge rate.<br>000: +2 Edge rate.<br>111: +1 Edge rate.<br>110: Nominal edge rate.<br>101: -1 Edge rate.<br>100: -2 Edge rate (fastest). | 0x1 |
| PICMG_REDUCED_POWER_ENA | 12 | R/W | Enable PICMC reduce power mode: In this mode, portions of the DSP processor are turned off, which reduces the PHY's operating power. The DSP performance characteristics in this mode are configured to support the channel characteristics specified in the PICMC 2.16 and PICMC 3.0 specifications. The application of this mode is in environments that have a high signal to noise ratio on the media. For example, Ethernet over backplane, or where cable length is short (less than 10m). When this field is cleared, the PHY operates in normal DSP mode. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| RESERVED | 8:6 | R/W | Must be set to its default. | 0x1 |

*Table 786 •* **Fields in PHY_CTRL_EXT2 *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| JUMBO_PKT_ENA | 5:4 | R/W | Controls the symbol buffering for the receive synchronization FIFO used in 1000BASE-T mode. Other encodings are reserved and must not be selected. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>Note: When set, the default maximum packet values are based on 100 ppm driven reference clock to the device. Controlling the ppm offset between the MAC and the PHY as specified in the field encoding description results in a higher jumbo packet length.<br>00: Normal IEEE 1518-byte packet length.<br>01: 9-kilobyte jumbo packet length (12 kilobytes with 60 ppm or better reference clock).<br>10: 12-kilobyte jumbo packet length (16 kilobytes with 70 ppm or better reference clock).<br>11: Reserved. | 0x0 |
| RESERVED | 3:1 | R/W | Must be set to its default. | 0x0 |
| CON_LOOPBACK_1000BT_ENA | 0 | R/W | Set PHY into 1000BASE-T connector loopback mode. When enabled, the PHY only works with a connector loopback. | 0x0 |

### 7.20.1.24 PHY:PHY_STD:PHY_INT_MASK

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 787 •* **Fields in PHY_INT_MASK**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PHY_INT_ENA | 15 | R/W | Enable global PHY interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| SPEED_STATE_CHANGE_INT_ENA | 14 | R/W | Set to unmask speed change interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |

*Table 787 •* **Fields in PHY_INT_MASK *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LINK_STATE_CHANGE_INT_ENA | 13 | R/W | Set to unmask link state/energy detected change interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| FDX_STATE_CHANGE_INT_ENA | 12 | R/W | Set to unmask FDX change interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| AUTONEG_ERR_INT_ENA | 11 | R/W | Set to unmask auto-negotiation error interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| AUTONEG_DONE_INT_ENA | 10 | R/W | Set to unmask auto-negotiation-done/interlock done interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| INLINE_POW_DET_INT_ENA | 9 | R/W | Set to unmask In-line Powered Device Detected interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| SYMBOL_ERR_INT_ENA | 8 | R/W | Set to unmask Symbol Error interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| FAST_LINK_FAIL_INT_ENA | 7 | R/W | Set to unmask fast link failure interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| TX_FIFO_INT_ENA | 6 | R/W | Set to unmask TX FIFO interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| RX_FIFO_INT_ENA | 5 | R/W | Set to unmask RX FIFO interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| FALSE_CARRIER_INT_ENA | 3 | R/W | Set to unmask False Carrier interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| LINK_SPEED_DOWNSHIFT_INT_ENA | 2 | R/W | Set to unmask link speed downshift interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |

*Table 787 •* **Fields in PHY_INT_MASK *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| MASTER_SLAVE_INT_ENA | 1 | R/W | Set to unmask master/slave interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| RX_ER_INT_ENA | 0 | R/W | Set to unmask RX_ER interrupt. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |

### 7.20.1.25 PHY:PHY_STD:PHY_INT_STAT

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 788 •* **Fields in PHY_INT_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PHY_INT_PEND | 15 | R/O | Set when an unacknowledged 'global' PHY interrupt is pending, the cause of the interrupt can be determined by examining the other fields of this register. This field is set no matter the state of PHY_INT_MASK.PHY_INT_ENA. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| SPEED_STATE_CHANGE_INT_PEND | 14 | R/O | Set when a speed interrupt is pending, this is activated when the operating speed of the PHY changes (requires that auto-negotiation is enabled; see PHY_CTRL.AUTONEG_ENA). This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| LINK_STATE_CHANGE_INT_PEND | 13 | R/O | Set when a Link State/Energy Detected interrupt is pending. This interrupt occurs when the link status of the PHY changes, or if ActiPHY mode is enabled and energy is detected on the media (see PHY_AUX_CTRL_STAT.ACTIPHY_ENA). This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |

*Table 788 •* **Fields in PHY_INT_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FDX_STATE_CHANGE_INT_PEND | 12 | R/O | Set when an FDX interrupt is pending. FDX interrupt is caused when the FDX/HDX state of the PHY changes (requires that auto-negotiation is enabled; see PHY_CTRL.AUTONEG_ENA). This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| AUTONEG_ERR_INT_PEND | 11 | R/O | Set when an auto-negotiation Error interrupt is pending, this is caused when an error is detected by the auto-negotiation state machine. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| AUTONEG_DONE_INT_PEND | 10 | R/O | Set when an auto-negotiation-Done/Interlock Done interrupt is pending, this is caused when the Auto-negotiation finishes a negotiation process. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| INLINE_POW_DET_INT_PEND | 9 | R/O | Set when an In-line Powered Device Detected interrupt is pending. This interrupt is caused when a device requiring in-line power is detected (requires that detection is enabled; see PHY_CTRL_EXT4.INLINE_DETECT_ENA). This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| SYMBOL_ERR_INT_PEND | 8 | R/O | Set when a Symbol Error interrupt is pending, this is caused by detection of a symbol error by the descrambler. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| FAST_LINK_FAIL_INT_PEND | 7 | R/O | Set when a fast link failure interrupt is pending. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| TX_FIFO_INT_PEND | 6 | R/O | Set when a TX FIFO interrupt is pending. TX FIFO interrupt is generated by TX FIFO underflow or overflow. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |

*Table 788 •* **Fields in PHY_INT_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RX_FIFO_INT_PEND | 5 | R/O | Set when a RX FIFO interrupt is pending. This interrupt is caused by RX FIFO underflow or overflow. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| FALSE_CARRIER_INT_PEND | 3 | R/O | Set when a False Carrier interrupt is pending. False Carrier interrupt is generated when the PHY detects a false carrier. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| LINK_SPEED_DOWNSHIFT_INT_PEND | 2 | R/O | Set when a link speed downshift interrupt is pending. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| MASTER_SLAVE_ERR_INT_PEND | 1 | R/O | Set when a master/slave interrupt is pending. This interrupt is set when a master/slave resolution error us detected. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |
| RX_ER_INT_PEND | 0 | R/O | Set when a RX_ER interrupt is pending. This interrupt is set when an RX_ER condition occurs. This field is self-clearing; i.e. interrupt is acknowledged on read. | 0x0 |

### 7.20.1.26 PHY:PHY_STD:PHY_AUX_CTRL_STAT

**Parent:** PHY:PHY_STD

**Instances:** 1

Copied fields have the same default values as their source fields.

*Table 789 •* **Fields in PHY_AUX_CTRL_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| AUTONEG_COMPLETE_AUX | 15 | R/O | A read-only copy of PHY_STAT.AUTONEG_COMPLETE. Repeated here for convenience. See note for this register. | 0x0 |
| AUTONEG_STAT | 14 | R/O | When set the auto-negotiation function has been disabled (in PHY_CTRL.AUTONEG_ENA.) | 0x0 |

*Table 789 •* **Fields in PHY_AUX_CTRL_STAT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NO_MDI_X_IND | 13 | R/O | When this field is set, the auto-negotiation state machine has determined that crossover does not exist in the signal path. This field is only valid after 'descrambler lock' has been achieved (see PHY_STAT_1000BT_EXT.DESCRAM_LOCKED) and 'automatic pair swap correction' is enabled (see PHY_BYPASS_CTRL.PAIR_SWAP_DISABLE). | 0x0 |
| CD_PAIR_SWAP | 12 | R/O | When this field is set, the PHY has determined that the subchannel cable pairs C and D were swapped between the far-end transmitter and the receiver. In this case, the PHY corrects this internally. This field is only valid when auto-negotiation is complete (see AUTONEG_COMPLETE). | 0x0 |
| A_POL_INVERSION | 11 | R/O | When set, this field indicates that the polarity of pair A was inverted between the far-end transmitter and the receiver. In this case the PHY corrects this internally. This field is only valid when auto-negotiation is complete (see AUTONEG_COMPLETE).<br>0: Polarity is swapped on pair A.<br>1: Polarity is not swapped on pair A. | 0x0 |
| B_POL_INVERSION | 10 | R/O | When set, this field indicates that the polarity of pair B was inverted between the far-end transmitter and the receiver. In this case the PHY corrects this internally. This field is only valid when auto-negotiation is complete (see AUTONEG_COMPLETE).<br>0: Polarity is swapped on pair B.<br>1: Polarity is not swapped on pair B. | 0x0 |

*Table 789 •* **Fields in PHY_AUX_CTRL_STAT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| C_POL_INVERSION | 9 | R/O | When set, this field indicates that the polarity of pair C was inverted between the far-end transmitter and the receiver. In this case the PHY corrects this internally. This field is only valid when auto-negotiation is complete (see AUTONEG_COMPLETE) and in 1000BASE-T mode.<br>0: Polarity is swapped on pair C.<br>1: Polarity is not swapped on pair C. | 0x0 |
| D_POL_INVERSION | 8 | R/O | When set, this field indicates that the polarity of pair D was inverted between the far-end transmitter and the receiver. In this case the PHY corrects this internally. This field is only valid when auto-negotiation is complete (see AUTONEG_COMPLETE) and in 1000BASE-T mode.<br>0: Polarity is swapped on pair D.<br>1: Polarity is not swapped on pair D. | 0x0 |
| ACTIPHY_LINK_TIMER_MSB_CFG | 7 | R/W | Most significant bit of the link status time-out timer. Together with ACTIPHY_LINK_TIMER_LSB_CFG, this field determines the duration from losing the link to the ActiPHY enters low power state.<br>0: 1 seconds.<br>1: 2 seconds.<br>2: 3 seconds.<br>3: 4 seconds. | 0x0 |
| ACTIPHY_ENA | 6 | R/W | Enable ActiPHY power management mode. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| FDX_STAT | 5 | R/O | This field indicates the actual FDX/HDX operating mode of the PHY.<br>0: Half-duplex.<br>1: Full-duplex. | 0x0 |
| SPEED_STAT | 4:3 | R/O | This field indicates the actual operating speed of the PHY.<br>0: Speed is 10BASE-T.<br>1: Speed is 100BASE-TX.<br>2: Speed is 1000-BASE-T.<br>3: Reserved. | 0x0 |

*Table 789 •* **Fields in PHY_AUX_CTRL_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| ACTIPHY_LINK_TIMER_LSB_CFG | 2 | R/W | See ACTIPHY_LINK_TIMER_MSB_CFG. | 0x1 |

### 7.20.1.27 PHY:PHY_STD:PHY_LED_MODE_SEL

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 790 •* **Fields in PHY_LED_MODE_SEL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 15:12 | R/W | Must be set to its default. | 0x8 |
| RESERVED | 11:8 | R/W | Must be set to its default. | 0x0 |
| LED1_MODE_SEL | 7:4 | R/W | Select from LED modes 0 through 15. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x2 |
| LED0_MODE_SEL | 3:0 | R/W | Select from LED modes 0 through 15. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>0: Link/Activity<br>1: Link1000/Activity<br>2: Link100/Activity<br>3: Link10/Activity<br>4: Link100/1000/Activity<br>5: Link10/1000/Activity<br>6: Link10/100/Activity<br>7: Reserved<br>8: Duplex/Collision<br>9: Collision<br>10: Activity<br>11: Reserved<br>12: Auto-Negotiation Fault<br>13: Reserved<br>14: Force LED Off<br>15: Force LED On | 0x1 |

### 7.20.1.28 PHY:PHY_STD:PHY_LED_BEHAVIOR_CTRL

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 791 •* **Fields in PHY_LED_BEHAVIOR_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PULSING_ENA | 12 | R/W | Enable LED pulsing with programmable duty cycle. The duty cycle is programmed in PHY_GP::PHY_ENHANCED_LED_CTRL.LED_PULSE_DUTY. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>0: Normal operation.<br>1: LEDs pulse with a 5 kHz, programmable duty cycle when active. | 0x0 |
| BLINK_RATE_CFG | 11:10 | R/W | Configure blink rate of LEDs when applicable. If pulse stretching has been selected rather than blink, this controls the stretch-period rather than frequency. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>00: 2.5 Hz blink rate/400 ms pulse-stretch.<br>01: 5 Hz blink rate/200 ms pulse-stretch.<br>10: 10 Hz blink rate/100 ms pulse-stretch.<br>11: 20 Hz blink rate/50 ms pulse-stretch. The blink rate selection for PHY0 globally sets the rate used for all LED pins on all PHY ports. | 0x1 |
| LED1_PULSE_STRETCH_ENA | 6 | R/W | Enable pulse-stretch behavior instead of blinking for LED1. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| LED0_PULSE_STRETCH_ENA | 5 | R/W | Enable pulse-stretch behavior instead of blinking for LED0. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| LED1_COMBINE_DIS | 1 | R/W | Disabling of the LED1 combine feature. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>0: Combine enabled (link/activity, duplex/collision).<br>1: Disable combination (link only, duplex only). | 0x0 |

*Table 791 •* **Fields in PHY_LED_BEHAVIOR_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LED0_COMBINE_DIS | 0 | R/W | Disabling of the LED0 combine feature. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>0: Combine enabled (link/activity, duplex/collision).<br>1: Disable combination (link only, duplex only). | 0x0 |

### 7.20.1.29 PHY:PHY_STD:PHY_MEMORY_PAGE_ACCESS

**Parent:** PHY:PHY_STD

**Instances:** 1

*Table 792 •* **Fields in PHY_MEMORY_PAGE_ACCESS**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PAGE_ACCESS_CFG | 4:0 | R/W | This bit controls the mapping of PHY registers 0x10 through 0x1E. When changing pages, all registers in the range 0x10 through 0x1E are replaced - even if the new memory-page does not define all addresses in the range 0x10 through 0x1E.<br>0: Register Page 0 is mapped (standard set).<br>1: Register Page 1 is mapped (extended set 1).<br>2: Register Page 2 is mapped (extended set 2).<br>16: Register Page 16 is mapped (general purpose). | 0x00 |

## 7.20.2 PHY:PHY_EXT1

**Parent:** PHY

**Instances:** 1

Set register 0x1F to 0x0001 to make the extended set of registers visible in the address range 0x10 through 0x1E. Set register 0x1F to 0x0000 to revert back to the standard register set.

*Table 793 •* **Registers in PHY_EXT1**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_CRC_GOOD_CNT | 0x00000012 | 1 | CRC Good Counter (Address 18E1) | Page 591 |

*Table 793 •* **Registers in PHY_EXT1 *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_EXT_MODE_CTRL | 0x00000013 | 1 | Extended Mode Control (Address 19E1) | Page 591 |
| PHY_CTRL_EXT3 | 0x00000014 | 1 | Extended Control Number 3 (Address 20E1) | Page 592 |
| PHY_CTRL_EXT4 | 0x00000017 | 1 | Extended Control Number 4 (Address 23E1) | Page 593 |
| PHY_1000BT_EPG1 | 0x0000001D | 1 | 1000BASE-T Ethernet Packet Generator Number 1 (Address 29E1) | Page 594 |
| PHY_1000BT_EPG2 | 0x0000001E | 1 | 1000BASE-T Ethernet Packet Generator Number 2 (Address 30E1) | Page 596 |

### 7.20.2.1 PHY:PHY_EXT1:PHY_CRC_GOOD_CNT

**Parent:** PHY:PHY_EXT1

**Instances:** 1

*Table 794 •* **Fields in PHY_CRC_GOOD_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PACKET_SINCE_LAST_READ | 15 | R/O | Packet received since last read. This is a self-clearing bit. | 0x0 |
| CRC_GOOD_PKT_CNT | 13:0 | R/O | Counter containing the number of packets with valid CRCs; this counter does not saturate and rolls over. This is a self-clearing field. | 0x0000 |

### 7.20.2.2 PHY:PHY_EXT1:PHY_EXT_MODE_CTRL

**Parent:** PHY:PHY_EXT1

**Instances:** 1

*Table 795 •* **Fields in PHY_EXT_MODE_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LED1_EXT_MODE_ENA | 13 | R/W | Enable extended LED mode for LED1. For available LED modes, see LED0_EXT_MODE_ENA. | 0x0 |

*Table 795 •* **Fields in PHY_EXT_MODE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LED0_EXT_MODE_ENA | 12 | R/W | Enable extended LED mode for LED0. If set, the available LED modes selected in PHY_LED_MODE_SEL.LED0_MODE_SEL are:<br>0-3: Reserved<br>4: Force LED Off.<br>5: Force LED On. LED pulsing is disabled in this mode.<br>6: Fast Link Fail.<br>7-15: Reserved. | 0x0 |
| LED_BLINK_SUPPRESS | 11 | R/W | Suppress LED blink after reset.<br>0: Suppress LED blink after COMA_MODE is deasserted.<br>1: Blink LEDs after COMA_MODE is deasserted. | 0x0 |
| FORCE_MDI_CROSSOVER_ENA | 3:2 | R/W | Force MDI crossover.<br>00: Normal HP Auto-MDIX operation.<br>01: Reserved.<br>10: Copper media forced to MDI.<br>11: Copper media forced MDI-X. | 0x0 |

### 7.20.2.3 PHY:PHY_EXT1:PHY_CTRL_EXT3

**Parent:** PHY:PHY_EXT1

**Instances:** 1

*Table 796 •* **Fields in PHY_CTRL_EXT3**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 15 | R/W | Must be set to its default. | 0x1 |
| ACTIPHY_SLEEP_TIMER | 14:13 | R/W | This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>00: 1 second.<br>01: 2 seconds.<br>10: 3 seconds.<br>11: 4 seconds. | 0x1 |
| ACTIPHY_WAKEUP_TIMER | 12:11 | R/W | This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>00: 160 ms.<br>01: 400 ms.<br>10: 800 ms.<br>11: 2 seconds. | 0x0 |

*Table 796 •* **Fields in PHY_CTRL_EXT3** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| NO_PREAMBLE_10BT_ENA | 5 | R/W | If set, 10BASE-T asserts RX_DV indication when data is presented to the receiver even without a preamble preceding it.This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| SPEED_DOWNSHIFT_ENA | 4 | R/W | Enables automatic downshift the auto-negotiation advertisement to the next lower available speed after the number of failed 1000BASE-T auto-negotiation attempts specified in SPEED_DOWNSHIFT_CFG. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| SPEED_DOWNSHIFT_CFG | 3:2 | R/W | Configures the number of unsuccessful 1000BASE-T auto-negotiation attempts that are required before the auto-negotiation advertisement is downshifted to the next lower available speed. This field applies only if automatic downshift of speed is enabled (see SPEED_DOWNSHIFT_ENA). This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA.<br>00: Downshift after 2 failed attempts.<br>01: Downshift after 3 failed attempts.<br>10: Downshift after 4 failed attempts.<br>11: Downshift after 5 failed attempts. | 0x1 |
| SPEED_DOWNSHIFT_STAT | 1 | R/O | This status field indicates that a downshift is required in order for link to be established. If automatic downshifting is enabled (see SPEED_DOWNSHIFT_ENA), the current link speed is a result of a downshift. | 0x0 |

### 7.20.2.4 PHY:PHY_EXT1:PHY_CTRL_EXT4

**Parent:** PHY:PHY_EXT1

**Instances:** 1

The reset value of the address fields (PHY_ADDR) corresponds to the PHY in which it resides.

*Table 797 •* **Fields in PHY_CTRL_EXT4**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PHY_ADDR | 15:11 | R/O | This field contains the PHY address of the current PHY port. | 0x00 |
| INLINE_POW_DET_ENA | 10 | R/W | Enables detection of inline powered device as part of the auto-negotiation process. This is a sticky field; see PHY_CTRL_STAT_EXT.STICKY_RESET_ENA. | 0x0 |
| INLINE_POW_DET_STAT | 9:8 | R/O | This field shows the status if a device is connected to the PHY that requires inline power. This field is only valid if inline powered device detection is enabled (see INLINE_POW_DET_ENA).<br>00: Searching for devices.<br>01: Device found that requires inline power.<br>10: Device found that does not require inline power.<br>11: Reserved. | 0x0 |
| CRC_1000BT_CNT | 7:0 | R/O | This field indicates how many packets are received that contain a CRC error. This field is cleared on read and saturates at all ones. | 0x00 |

### 7.20.2.5 PHY:PHY_EXT1:PHY_1000BT_EPG1

**Parent:** PHY:PHY_EXT1

**Instances:** 1

*Table 798 •* **Fields in PHY_1000BT_EPG1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EPG_ENA | 15 | R/W | Enables the Ethernet packet generator. When this field is set, the EPG is selected as the driving source for the PHY transmit signals, and the MAC transmit pins are disabled. | 0x0 |
| EPG_RUN_ENA | 14 | R/W | Begin transmission of Ethernet packets. Clear to stop the transmission of packets. If a transmission is in progress, the transmission of packets is stopped after the current packet is transmitted. This field is valid only when the EPG is enabled (see EPG_ENA). | 0x0 |

*Table 798 •* **Fields in PHY_1000BT_EPG1** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TRANSMIT_DURATION_CFG | 13 | R/W | Configure the duration of the packet generation. When set, the EPG continuously transmits packets as long as field EPG_RUN_ENA is set. When cleared, the EPG transmits 30,000,000 packets when field EPG_RUN_ENA is set, after which time, field EPG_RUN_ENA is automatically cleared. This field is latched when packet generation begins by setting EPG_RUN_ENA in this register. | 0x0 |
| PACKET_LEN_CFG | 12:11 | R/W | This field selects the length of packets to be generated by the EPG. This field is latched when generation of packets begins by setting EPG_RUN_ENA in this register.<br>00: 125-byte packets.<br>01: 64-byte packets.<br>10: 1518-byte packets.<br>11: 10,000-byte packets. | 0x0 |
| INTER_PACKET_GAB_CFG | 10 | R/W | This field configures the inter packet gab for packets generated by the EPG. This field is latched when generation of packets begins by setting EPG_RUN_ENA in this register.<br>0: 96 ns inter-packet gap.<br>1: 9,192 ns inter-packet gap. | 0x0 |
| DEST_ADDR_CFG | 9:6 | R/W | This field configures the low nibble of the most significant byte of the destination MAC address. The rest of the destination MAC address is all-ones. For example, setting this field to 0x2 results in packets generated with a destination MAC address of 0xF2FFFFFFFFFF. This field is latched when generation of packets begins by setting EPG_RUN_ENA in this register. | 0x1 |

*Table 798 •* **Fields in PHY_1000BT_EPG1 *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SRC_ADDR_CFG | 5:2 | R/W | This field configures the low nibble of the most significant byte of the source MAC address. The rest of the source MAC address is all-ones. For example, setting this field to 0xE results in packets generated with a source MAC address of 0xFEFFFFFFFFFF. This field is latched when generation of packets begins by setting EPG_RUN_ENA in this register. | 0x0 |
| PAYLOAD_TYPE | 1 | R/W | Payload type.<br>0: Fixed based on payload pattern.<br>1: Randomly generated payload pattern. | 0x0 |
| BAD_FCS_ENA | 0 | R/W | When this field is set, the EPG generates packets containing an invalid Frame Check Sequence (FCS). When cleared, the EPG generates packets with a valid FCS. This field is latched when generation of packets begins by setting EPG_RUN_ENA in this register. | 0x0 |

### 7.20.2.6 PHY:PHY_EXT1:PHY_1000BT_EPG2

**Parent:** PHY:PHY_EXT1

**Instances:** 1

*Table 799 •* **Fields in PHY_1000BT_EPG2**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PACKET_PAYLOAD_CFG | 15:0 | R/W | Each packet generated by the EPG contains a repeating sequence of this field as payload. This field is latched when generation of packets begins by setting PHY_1000BT_EPG1.EPG_RUN_ENA. | 0x0000 |

## 7.20.3 PHY:PHY_EXT2

**Parent:** PHY

**Instances:** 1

Set register 0x1F to 0x0002 to make the extended set of registers visible in the address range 0x10 through 0x1E. Set register 0x1F to 0x0000 to revert back to the standard register set.

*Table 800* • **Registers in PHY_EXT2**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_PMD_TX_CTRL | 0x00000010 | 1 | Cu PMD Transmit Control (Address 16E2) | Page 597 |
| PHY_EEE_CTRL | 0x00000011 | 1 | EEE and LED Control (Address 17E2) | Page 597 |

### 7.20.3.1 PHY:PHY_EXT2:PHY_PMD_TX_CTRL

**Parent:** PHY:PHY_EXT2

**Instances:** 1

This register consists of the bits that provide control over the amplitude settings for the transmit side Cu PMD interface. These bits provide the ability to make small adjustments in the signal amplitude to compensate for minor variations in the magnetic from different vendors. Extreme caution must be exercised when changing these settings from the default values as they have a direct impact on the signal quality. Changing these settings also affects the linearity and harmonic distortion of the transmitted signals. Please contact the Vitesse Applications Support team for further help with changing these values.

*Table 801* • **Fields in PHY_PMD_TX_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| SIG_AMPL_1000BT | 15:12 | R/W | 1000BT signal amplitude trim. | 0x0 |
| SIG_AMPL_100BTX | 11:8 | R/W | 100BASE-TX signal amplitude trim. | 0x2 |
| SIG_AMPL_10BT | 7:4 | R/W | 10BASE-T signal amplitude trim. | 0xF |
| SIG_AMPL_10BTE | 3:0 | R/W | 10BASE-Te signal amplitude trim. | 0x0 |

### 7.20.3.2 PHY:PHY_EXT2:PHY_EEE_CTRL

**Parent:** PHY:PHY_EXT2

**Instances:** 1

*Table 802* • **Fields in PHY_EEE_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_10BTE_ENA | 15 | R/W | Enable energy efficient (IEEE 802.3az) 10BASE-Te operating mode. | 0x0 |

*Table 802 •* **Fields in PHY_EEE_CTRL *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| FORCE_1000BT_ENA | 5 | R/W | Enable 1000BT force mode to allow PHY to link up in 1000BT mode without forcing master/slave when PHY_STD::PHY_CTRL.SPEED_SEL_LSB_CFG=0 and PHY_STD::PHY_CTRL.SPEED_SEL_MSB_CFG=1. | 0x0 |
| FORCE_LPI_TX_ENA | 4 | R/W | Force transmit LPI.<br>0: Transmit idles being received from the MAC.<br>1: Enable the EPG to transmit LPI on the MDI instead of normal idles when receiving normal idles from the MAC. | 0x0 |
| EEE_LPI_TX_100BTX_DIS | 3 | R/W | Disable transmission of EEE LPI on transmit path MDI in 100BASE-TX mode when receiving LPI from MAC. | 0x0 |
| EEE_LPI_RX_100BTX_DIS | 2 | R/W | Disable transmission of EEE LPI on receive path MAC interface in 100BASE-TX mode when receiving LPI from the MDI. | 0x0 |
| EEE_LPI_TX_1000BT_DIS | 1 | R/W | Disable transmission of EEE LPI on transmit path MDI in 1000BT mode when receiving LPI from MAC. | 0x0 |
| EEE_LPI_RX_1000BT_DIS | 0 | R/W | Disable transmission of EEE LPI on receive path MAC interface in 1000BT mode when receiving LPI from the MDI. | 0x0 |

## 7.20.4 PHY:PHY_GP

**Parent:** PHY

**Instances:** 1

Set register 0x1F to 0x0010 to access the general purpose registers. This sets all 32 registers to the general purpose register space. Set register 0x1F to 0x0000 to revert back to the standard register set.

*Table 803 •* **Registers in PHY_GP**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_COMA_MODE_CTRL | 0x0000000E | 1 | Coma Mode Control (Address 14G) | Page 599 |
| PHY_ENHANCED_LED_CTRL | 0x00000019 | 1 | Enhanced LED Control (Address 25G) | Page 599 |

*Table 803 •* **Registers in PHY_GP *(continued)***

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_GLOBAL_INT_STAT | 0x0000001D | 1 | Global Interrupt Status (Address 29G) | Page 600 |

### 7.20.4.1 PHY:PHY_GP:PHY_COMA_MODE_CTRL

**Parent:** PHY:PHY_GP

**Instances:** 1

*Table 804 •* **Fields in PHY_COMA_MODE_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| COMA_MODE_OE | 13 | R/W | COMA_MODE output enable. Active low.<br>0: COMA_MODE pin is an output.<br>1: COMA_MODE pin is an input. | 0x1 |
| COMA_MODE_OUTPUT | 12 | R/W | COMA_MODE output data. | 0x0 |
| COMA_MODE_INPUT | 11 | R/O | COMA_MODE input data. | 0x0 |
| LED_TRISTATE_ENA | 9 | R/W | Tri-state enable for LEDs.<br>0: Drive LED bus output signals to high and low values as appropriate.<br>1: Tri-state LED output signals instead of driving them high. This allows those signals to be pulled above VDDIO using an external pull-up resistor. | 0x0 |

### 7.20.4.2 PHY:PHY_GP:PHY_ENHANCED_LED_CTRL

**Parent:** PHY:PHY_GP

**Instances:** 1

*Table 805 •* **Fields in PHY_ENHANCED_LED_CTRL**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| LED_PULSE_DUTY | 15:8 | R/W | LED pulsing duty cycle control. Programmable control for LED pulsing duty cycle when PHY_STD::PHY_LED_BEHAVIOR_CTRL.PULSING_ENA is set to 1. Valid settings are between 0 and 198. A setting of 0 corresponds to a 0.5% duty cycle and 198 corresponds to a 99.5% duty cycle. Intermediate values change the duty cycle in 0.5% increments. | 0x00 |

### 7.20.4.3 PHY:PHY_GP:PHY_GLOBAL_INT_STAT

**Parent:** PHY:PHY_GP

**Instances:** 1

*Table 806* • **Fields in PHY_GLOBAL_INT_STAT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| RESERVED | 12 | R/O | Must be set to its default. | 0x1 |
| PHY11_INT_SRC | 11 | R/O | Indicates that PHY11 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY11. | 0x1 |
| PHY10_INT_SRC | 10 | R/O | Indicates that PHY10 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY10. | 0x1 |
| PHY9_INT_SRC | 9 | R/O | Indicates that PHY9 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY9. | 0x1 |
| PHY8_INT_SRC | 8 | R/O | Indicates that PHY8 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY8. | 0x1 |
| PHY7_INT_SRC | 7 | R/O | Indicates that PHY7 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY7. | 0x1 |
| PHY6_INT_SRC | 6 | R/O | Indicates that PHY6 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY6. | 0x1 |
| PHY5_INT_SRC | 5 | R/O | Indicates that PHY5 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY5. | 0x1 |
| PHY4_INT_SRC | 4 | R/O | Indicates that PHY4 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY4. | 0x1 |

*Table 806 •* **Fields in PHY_GLOBAL_INT_STAT *(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| PHY3_INT_SRC | 3 | R/O | Indicates that PHY3 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY3. | 0x1 |
| PHY2_INT_SRC | 2 | R/O | Indicates that PHY2 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY2. | 0x1 |
| PHY1_INT_SRC | 1 | R/O | Indicates that PHY1 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY1. | 0x1 |
| PHY0_INT_SRC | 0 | R/O | Indicates that PHY0 is the source of the interrupt when this bit is cleared. This bit is set high when reading register PHY_INT_STAT in PHY0. | 0x1 |

## 7.20.5 PHY:PHY_EEE

**Parent:** PHY

**Instances:** 1

Access to these registers is through the IEEE standard registers MMD_ACCESS_CFG and MMD_ADDR_DATA.

*Table 807 •* **Registers in PHY_EEE**

| Register Name | Offset within Register Group | Instances and Address Spacing | Description | Details |
|---|---|---|---|---|
| PHY_PCS_STATUS1 | 0x00000000 | 1 | PCS Status 1 (Address 3.1) | Page 601 |
| PHY_EEE_CAPABILITIES | 0x00000001 | 1 | EEE Capabilities (Address 3.20) | Page 602 |
| PHY_EEE_WAKE_ERR_CNT | 0x00000002 | 1 | EEE Wake Error Counter (Address 3.22) | Page 602 |
| PHY_EEE_ADVERTISEMENT | 0x00000003 | 1 | EEE Advertisement (Address 7.60) | Page 603 |
| PHY_EEE_LP_ADVERTISEMENT | 0x00000004 | 1 | EEE Link Partner Advertisement (Address 7.61) | Page 603 |

### 7.20.5.1 PHY:PHY_EEE:PHY_PCS_STATUS1

**Parent:** PHY:PHY_EEE

**Instances:** 1

Status of the EEE operation from the PCS for the link that is currently active.

*Table 808* • **Fields in PHY_PCS_STATUS1**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| TX_LPI_RECV | 11 | R/O | 0: LPI not received<br>1: Tx PCS has received LPI | 0x0 |
| RX_LPI_RECV | 10 | R/O | 1: Rx PCS has received LPI<br>0: LPI not received | 0x0 |
| TX_LPI_INDICATION | 9 | R/O | 1: Tx PCS is currently receiving LPI<br>0: PCS is not currently receiving LPI | 0x0 |
| RX_LPI_INDICATION | 8 | R/O | 1: Rx PCS is currently receiving LPI<br>0: PCS is not currently receiving LPI | 0x0 |
| PCS_RECV_LINK_STAT | 2 | R/O | 1: PCS receive link up<br>0: PCS receive link down | 0x0 |

### 7.20.5.2 PHY:PHY_EEE:PHY_EEE_CAPABILITIES

**Parent:** PHY:PHY_EEE

**Instances:** 1

Indicate the capability of the PCS to support EEE functions for each PHY type.

*Table 809* • **Fields in PHY_EEE_CAPABILITIES**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_1000BT | 2 | R/O | Set if EEE is supported for 1000BASE-T.<br>1: EEE is supported for 1000BASE-T<br>0: EEE is not supported for 1000BASE-T | 0x1 |
| EEE_100BTX | 1 | R/O | Set if EEE is supported for 100BASE-TX.<br>1: EEE is supported for 100BASE-TX<br>0: EEE is not supported for 100BASE-TX | 0x1 |

### 7.20.5.3 PHY:PHY_EEE:PHY_EEE_WAKE_ERR_CNT

**Parent:** PHY:PHY_EEE

**Instances:** 1

This register is used by PHY types that support EEE to count wake time faults where the PHY fails to complete its normal wake sequence within the time required for the specific PHY type. The definition of the fault event to be counted is defined for each PHY and can occur during a refresh or a wakeup as defined by the PHY. This 16-bit counter is reset to all zeros when the EEE wake error counter is read or when the PHY undergoes hardware or software reset.

*Table 810 •* **Fields in PHY_EEE_WAKE_ERR_CNT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_WAKE_ERR_CNT | 15:0 | R/O | Count of wake time faults for a PHY. | 0x0000 |

### 7.20.5.4 PHY:PHY_EEE:PHY_EEE_ADVERTISEMENT

**Parent:** PHY:PHY_EEE

**Instances:** 1

Defines the EEE advertisement that is sent in the unformatted next page following a EEE technology message code.

*Table 811 •* **Fields in PHY_EEE_ADVERTISEMENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_1000BT_ADV | 2 | R/W | Set if EEE is supported for 1000BASE-T.<br>1: Advertise that the 1000BASE-T has EEE capability.<br>0: Do not advertise that the 1000BASE-T has EEE capability. | 0x0 |
| EEE_100BTX_ADV | 1 | R/W | Set if EEE is supported for 100BASE-TX.<br>1: Advertise that the 100BASE-TX has EEE capability<br>0: Do not advertise that the 100BASE-TX has EEE capability | 0x0 |

### 7.20.5.5 PHY:PHY_EEE:PHY_EEE_LP_ADVERTISEMENT

**Parent:** PHY:PHY_EEE

**Instances:** 1

All the bits in the EEE LP Advertisement register are read only. A write to the EEE LP advertisement register has no effect. When the AN process has been completed, this register will reflect the contents of the link partner's EEE advertisement register.

*Table 812 •* **Fields in PHY_EEE_LP_ADVERTISEMENT**

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_1000BT_LP_ADV | 2 | R/O | Set if EEE is supported for 1000BASE-T by link partner.<br>1: Link partner is advertising EEE capability for 1000BASE-T<br>0: Link partner is not advertising EEE capability for 1000BASE-T | 0x0 |

*Table 812 •* **Fields in PHY_EEE_LP_ADVERTISEMENT** ***(continued)***

| Field Name | Bit | Access | Description | Default |
|---|---|---|---|---|
| EEE_100BTX_LP_ADV | 1 | R/O | Set if EEE is supported for 100BASE-TX by link partner.<br>1: Link partner is advertising EEE capability for 100BASE-TX<br>0: Link partner is not advertising EEE capability for 100BASE-TX | 0x0 |

# 8 Electrical Specifications

This section provides the DC characteristics, AC characteristics, recommended operating conditions, and stress ratings for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

## 8.1 DC Characteristics

This section contains the DC specifications for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

### 8.1.1 Internal Pull-Up or Pull-Down Resistors

Internal pull-up or pull-down resistors are specified in the following table. For more information about signals with internal pull-up or pull-down resistors, see Pins by Function for VSC7424-02, page 638, Pins by Function for VSC7425-02, page 661, Pins by Function for VSC7426-02, page 684, or Pins by Function for VSC7427-02, page 707.

All internal pull-up resistors are connected to their respective I/O supply.

***Table 813 •*** **Internal Pull-Up or Pull-Down Resistors**

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| Internal pull-up resistor, GPIO and SI pins | $R_{PU}$ | 33 | 53 | 90 | kΩ |
| Internal pull-up resistor, all other pins | $R_{PD}$ | 96 | 120 | 144 | kΩ |
| Internal pull-down resistor | $R_{PD}$ | 96 | 120 | 144 | kΩ |

### 8.1.2 Reference Clock

The following table lists the DC specifications for the differential RefClk signal. Differential and single-ended modes are supported. For more information about single-ended mode operation, see Single-Ended RefClk Input, page 733.

***Table 814 •*** **Reference Clock Input DC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit |
|---|---|---|---|---|
| Input voltage range | $V_{IP}$, $V_{IN}$ | –25 | 1260 | mV |
| Input differential voltage, peak-to-peak | $\|V_{ID}\|$ | 150(1) | 1000 | mV |
| Input common-mode voltage | $V_{CM}$ | 0 | 1200(2) | mV |

1. To meet jitter specifications, the minimum input differential voltage must be 400 mV. When using a single-ended clock input, the RefClk_P low voltage level must be lower than $V_{DD_A}$ – 200 mV, and the high voltage level must be higher than $V_{DD_A}$ + 200 mV.
2. The maximum common-mode voltage is given without any differential signal, because the input is internally AC-coupled. The common-mode voltage is only limited by the maximum and minimum input voltage range and the input signal's differential amplitude.

### 8.1.3 DDR2 SDRAM Interface

The DDR2 SDRAM interface supports the requirements of SDRAM devices as described in the JEDEC DDR2 specifications. The SDRAM interface signals are compatible with JESD79-2E (DDR2 SDRAM Specification, April 2008) and the JESD8-15A (Stub Series Terminated Logic for 1.8V (SSTL_18), September 2003). The SSTL I/O buffers have programmable on-die termination (ODT).

The following table lists the DC specifications for SDRAM interface signals.

***Table 815 •*** **DDR2 SDRAM Signal DC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Input voltage reference(1) | DDR_$V_{REF}$ | 49% $V_{DD_IODDR}$ | 51% $V_{DD_IODDR}$ | V | |
| Input voltage high | $V_{IH(DC)}$ | DDR_$V_{REF}$ + 0.125 | $V_{DD_IODDR}$ + 0.3 | V | |
| Input voltage low | $V_{IL(DC)}$ | –0.3 | DDR_$V_{REF}$ – 0.125 | V | |
| Input leakage current | $\|I_L\|$ | | 58 | µA | $0V \le V_I \le V_{DD_IODDR}$. |
| Output source DC current(2) | $I_{OH}$ | –6 | | mA | External 50 Ω termination to $V_{DD_IODDR}/2$. |
| Output sink DC current(2) | $I_{OL}$ | 6 | | mA | External 50 Ω termination to $V_{DD_IODDR}/2$. |

1. DDR_$V_{REF}$ is expected to track variations in $V_{DD_IODDR}$. Peak-to-peak AC noise on DDR_$V_{REF}$ must not exceed ±2% of DDR_$V_{REF}$.
2. With 40 Ω output driver impedance.

## 8.1.4 SGMII DC Definitions and Test Circuits

This section provides information about the definitions and test circuits that apply to certain parameters for the Enhanced SerDes and SerDes interfaces. The following illustrations show the DC definitions for the SGMII inputs and outputs.

***Figure 79 •*** **SGMII DC Input Definitions**

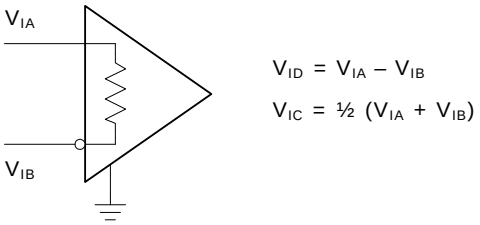



***Figure 80 •*** **SGMII DC Transmit Test Circuit**

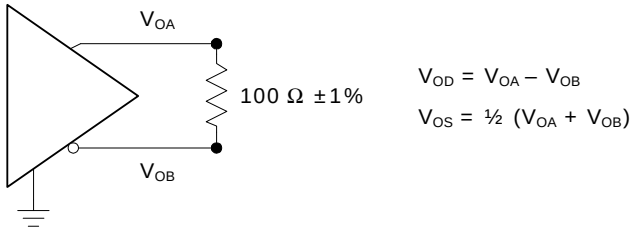
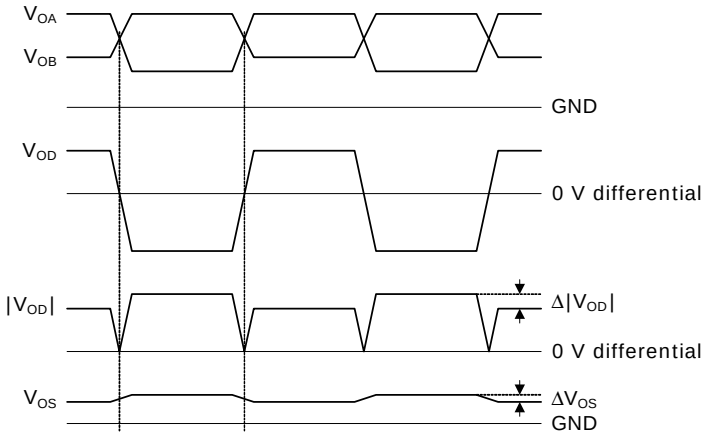


***Figure 81 •*** **SGMII DC Definitions**

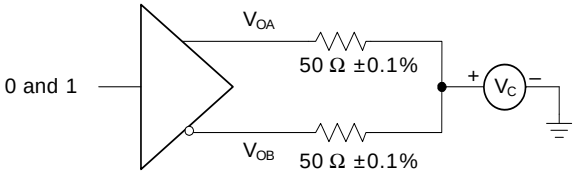



$\Delta|V_{OD}| = | |V_{OAH} - V_{OBL}| - |V_{OBH} - V_{OAL}| |$

$\Delta V_{OS} = | ½(V_{OAH} + V_{OBL}) - ½(V_{OAL} + V_{OBH}) |$

The following illustrations show the SMGII DC driver output impedance test circuit and the DC input definitions.

***Figure 82 •*** **SGMII DC Driver Output Impedance Test Circuit**




## 8.1.5 Enhanced SerDes Interface

All DC specifications for the Enhanced SerDes interface are compliant with the QSGMII Specification Revision 1.3 and meet or exceed the requirements in the standard. They are also compliant with the OIF-CEI version 2.0 requirements where applicable.

The Enhanced SerDes interface supports three major modes: SGMII, QSGMII, and SFP. The values in the following table apply to modes specified.

***Table 816 •*** **Enhanced SerDes Driver DC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Output differential peak voltage(1), 1.0 V, SFP and QSGMII modes | $\|V_{ODp}\|$ | 250 | 400 | mV | $V_{DD_VS}$ = 1.0 V. $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.0 V and 1.2 V, SGMII mode | $\|V_{ODp}\|$ | 150 | 400 | mV | $V_{DD_VS}$ = 1.0 V, $V_{DD_VS}$ = 1.2 V. $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.2 V, SFP mode | $\|V_{ODp}\|$ | 300 | 600 | mV | $V_{DD_VS}$ = 1.2 V. $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.2 V, QSGMII mode | $\|V_{ODp}\|$ | 200 | 400 | mV | $V_{DD_VS}$ = 1.2 V. $R_L$ = 100 Ω ±1%. |

*Table 816 •* **Enhanced SerDes Driver DC Specifications *(continued)***

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| DC output impedance, single-ended, SGMII mode | $R_O$ | 40 | 140 | Ω | $V_C$ = 1.0 V and 1.2 V. See Figure 82, page 607. |
| $R_O$ mismatch between A and B(2), SGMII mode | $\Delta R_O$ | | 10 | % | $V_C$ = 1.0 V and 1.2 V. See Figure 82, page 607. |
| Change in $\|V_{OD}\|$ between 0 and 1, SGMII mode | $\Delta\|V_{OD}\|$ | | 25 | mV | $R_L$ = 100 Ω ±1% |
| Change in $V_{OS}$ between 0 and 1, SGMII mode | $\Delta V_{OS}$ | | 25 | mV | $R_L$ = 100 Ω ±1%. |
| Output current, driver shorted to GND, SGMII and QSGMII modes | $\|I_{OSA}\|$, $\|I_{OSB}\|$ | | 40 | mA | |
| Output current, drivers shorted together, SGMII and QSGMII modes | $\|I_{OSAB}\|$ | | 12 | mA | |

1. Voltage is adjustable in 64 steps. For more information about setting the adjustable voltages, see the OB_LEV bit in Table 520, page 407. To meet the return loss specification, the maximum swing is 600 mV peak-to-peak for $V_{DD_VS}$ = 1.0 V and 950 mV peak-to-peak for $V_{DD_VS}$ = 1.2 V.
2. Matching of reflection coefficients. For more information about test methods, see IEEE 1596.3-1996.

The following table lists the DC specifications for the Enhanced SerDes receivers. In most applications, AC-coupling is required. For more information, see Enhanced SerDes Interface, page 735.

*Table 817 •* **Enhanced SerDes Receiver DC Specifications**

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| Input voltage range, $V_{IA}$ or $V_{IB}$(1) | $V_I$ | –0.25 | | 1.2 | V |
| Input differential peak voltage(2), SGMII and SFP modes | $\|V_{ID}\|$ | 50 | | 800 | mV |
| Input differential peak voltage(2), QSGMII mode | $\|V_{ID}\|$ | 50 | | 600 | mV |
| Receiver differential input impedance | $R_I$ | 80 | 100 | 120 | Ω |

1. QSGMII DC input sensitivity is <400 mV.
2. Ranges specified are for optimal operation.

## 8.1.6 SerDes (SGMII) Interface

The SerDes output drivers are designed to operate in an SGMII/LVDS mode and in a high-drive/PECL mode (SFP and 1000BASE-KX modes). The SGMII/LVDS mode meets or exceeds the DC requirements of the Serial-GMII Specification version 1.9, unless otherwise noted.

The following table lists the DC specifications for the SGMII driver. The values are valid for all configurations, unless stated otherwise.

***Table 818 •*** **SerDes Driver DC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Output high voltage, $V_{OA}$ or $V_{OB}$ | $V_{OH}$ | | 1250 | mV | $R_L$ = 100 Ω ±1%. |
| Output low voltage, $V_{OA}$ or $V_{OB}$ | $V_{OL}$ | 0 | | mV | $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.0 V | $\|V_{OD}\|$ | 150 | 400 | mV | $V_{DD_VS}$ = 1.0 V, $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.2 V | $\|V_{OD}\|$ | 150 | 600 | mV | $V_{DD_VS}$ = 1.2 V, $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.0 V and 1.2 V, SGMII mode | $\|V_{OD}\|$ | 150 | 400 | mV | $V_{DD_VS}$ = 1.0 V, $V_{DD_VS}$ = 1.2 V. $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.2 V, 1000BASE-KX mode | $\|V_{OD}\|$ | 400 | 600 | mV | $V_{DD_VS}$ = 1.2 V, $R_L$ = 100 Ω ±1%. |
| Output differential peak voltage(1), 1.2 V, SFP mode | $\|V_{OD}\|$ | 300 | 600 | mV | $V_{DD_VS}$ = 1.2 V, $R_L$ = 100 Ω ±1%. |
| Output offset voltage(2), 1.0 V | $V_{OS}$ | 420 | 580 | mV | $V_{DD_VS}$ = 1.0 V, $R_L$ = 100 Ω ±1%. |
| Output offset voltage(2), 1.2 V | $V_{OS}$ | 445 | 605 | mV | $V_{DD_VS}$ = 1.2 V, $R_L$ = 100 Ω ±1%. |
| DC output impedance, single-ended, SMGII mode | $R_O$ | 40 | 140 | Ω | $V_C$ = 1.0 V and 1.2 V. See Figure 82, page 607. |
| $R_O$ mismatch between A and B(3), SMGII mode | $\Delta R_O$ | | 10 | % | $V_C$ = 1.0 V and 1.2 V. See Figure 82, page 607. |
| Change in $\|V_{OD}\|$ between 0 and 1, SGMII mode | $\Delta\|V_{OD}\|$ | | 25 | mV | $R_L$ = 100 Ω ±1%. |
| Change in $V_{OS}$ between 0 and 1, SGMII mode | $\Delta V_{OS}$ | | 25 | mV | $R_L$ = 100 Ω ±1%. |
| Output current, driver shorted to GND, SGMII mode | $\|I_{OSA}\|$, $\|I_{OSB}\|$ | | 40 | mA | |
| Output current, drivers shorted together, SGMII mode | $\|I_{OSAB}\|$ | | 12 | mA | |

1. Voltage is adjustable in 14 steps. For more information about setting the adjustable voltages, see the OB_AMP_CTRL bit in Table 505, page 399. To meet the return loss specification, the maximum swing is 600 mV peak-to-peak for $V_{DD_VS}$ = 1.0 V and 950 mV peak-to-peak for $V_{DD_VS}$ = 1.2 V.
2. Requires AC-coupling for SGMII compliance.
3. Matching of reflection coefficients. For more information about test methods, see IEEE 1596.3-1996.

The following table lists the DC specifications for the SGMII receivers.

***Table 819 •*** **SerDes Receiver DC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Input voltage range, $V_{IA}$ or $V_{IB}$ | $V_I$ | –25 | 1250 | mV | |
| Input differential peak voltage | $\|V_{ID}\|$ | 50 | 1000 | mV | |
| Input common-mode voltage(1) | $V_{IC}$ | 0 | $V_{DD_A}$(2) | mV | Without any differential signal (internally AC-coupled) |
| Receiver differential input impedance | $R_I$ | 80 | 120 | Ω | |
| Input differential hysteresis, SGMII mode | $V_{HYST}$ | 25 | | mV | |

1. SGMII compliancy requires external AC-coupling. When interfacing with specific Vitesse devices, DC-coupling is possible. For more information, contact your Vitesse representative.
2. The maximum common-mode voltage is given without any differential signal, because the input is internally AC-coupled. The common-mode voltage is only limited by the maximum and minimum input voltage range and the input signal's differential swing.

## 8.1.7 MIIM, GPIO, SI, JTAG, and Miscellaneous Signals

This section provides the DC specifications for the MII Management (MIIM), GPIO, SI, JTAG, and miscellaneous signals. The following I/O signals comply with the specifications provided in this section.

***Table 820 •***

| | | |
|---|---|---|
| MDC | JTAG_nTRST | Reserved |
| MDIO | JTAG_TMS | RefClk_Sel[2:0] |
| GPIO[31:0] | JTAG_TDO | VCORE_CFG[2:0] |
| SI_Clk | JTAG_TCK | VCore_ICE_nEN |
| SI_DI | JTAG_TDI | |
| SI_DO | nReset | |
| SI_nEn | COMA_MODE | |

The outputs and inputs meet or exceed the requirements of the LVTTL and LVCMOS standard, JEDEC JESD8-B (September 1999) standard, unless otherwise stated. The inputs are Schmitt-trigger for noise immunity.

***Table 821 •*** **MIIM, GPIO, SI, JTAG, and Miscellaneous Signals DC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Output high voltage, $I_{OH}$ = –12 mA | $V_{OH}$ | 1.7 | | V | |
| Output high voltage, $I_{OH}$ = –2 mA | $V_{OH}$ | 2.1 | | V | |
| Output low voltage, $I_{OL}$ = 12 mA | $V_{OL}$ | | 0.7 | V | |
| Output low voltage, $I_{OL}$ = 2 mA | $V_{OL}$ | | 0.4 | V | |
| Input high voltage | $V_{IH}$ | 1.85 | 3.6 | V | |
| Input low voltage | $V_{IL}$ | –0.3 | 0.8 | V | |
| Input high current(1) | $I_{IH}$ | | 10 | µA | $V_I = V_{DD_IO}$ |
| Input low current(1) | $I_{IL}$ | –10 | | µA | $V_I$ = 0 V |
| Input capacitance | $C_I$ | | 10 | pF | |

1. Input high current and input low current equals the maximum leakage current, excluding the current in the built-in pull resistors.

## 8.1.8 Thermal Diode

The VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices include an on-die diode and internal circuitry for monitoring die temperature (junction temperature). The operation and accuracy of the diode is not guaranteed and should only be used as a reference.

A thermal sensor, located on the board or in a stand-alone measurement kit, can monitor and display the die temperature of the switch for thermal management or instrumentation purposes.

Temperature measurement using a thermal diode is very sensitive to noise.

The following table provides the diode parameter and interface specifications. Note that the THERMDC_VSS pin is connected to VSS internally in the devices.

***Table 822 •*** **Thermal Diode Parameters**

| Parameter | Symbol | Typical | Maximum | Unit |
|---|---|---|---|---|
| Forward bias current | IFW | | 1 | mA |

*Table 822 •* **Thermal Diode Parameters**

| Parameter | Symbol | Typical | Maximum | Unit |
|---|---|---|---|---|
| Diode ideality factor | n | 1.008 | | |

**Notes** Vitesse does not support or recommend operation of the thermal diode under reverse bias.

The ideality factor, n, represents the deviation from ideal diode behavior as exemplified by the diode equation:

$$I_{FW} = I_S \times \left( e^{V_d \times \frac{q}{nkT}} - 1 \right)$$

where, $I_S$ = saturation current, q = electronic charge, Vd = voltage across the diode, k = Boltzmann Constant, and T = absolute temperature (Kelvin).

# 8.2 AC Characteristics

This section provides the AC specifications for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

## 8.2.1 Reference Clock

The signal applied to the RefClk differential input must comply with the requirements listed in the following table at the pin of the device.

To meet QSGMII jitter generation requirements, Microsemi requires the use of a differential reference clock source. Use of a 25 MHz single-ended reference clock is not recommended. However, to implement a QSGMII chip interconnect using a 25 MHz single-ended reference clock and achieve error-free data transfer on that interface, use an Ethernet PHY with higher jitter tolerance than specified in the standard, such as Microsemi's VSC8512-02 or VSC8522-02. For more information about QSGMII interoperability when using a 25 MHz single-ended reference clock, contact your Microsemi representative.

*Table 823 •* **Reference Clock AC Specifications**

| Parameter | Symbol | Minimum | Typical | Maximum | Unit | Condition |
|---|---|---|---|---|---|---|
| RefClk frequency, RefClk_Sel = 000 | *f* | –100 ppm | 125 | 100 ppm | MHz | |
| RefClk frequency, RefClk_Sel = 001 | *f* | –100 ppm | 156.25 | 100 ppm | MHz | |
| RefClk frequency, RefClk_Sel = 100 | *f* | –100 ppm | 25 | 100 ppm | MHz | |
| Clock duty cycle | | 40 | | 60 | % | Measured at 50% threshold. |
| Rise time and fall time | $t_R$, $t_F$ | | | 1.5 | ns | 20% to 80% threshold. |
| RefClk input RMS jitter, bandwidth between 12 kHz and 500 kHz | | | | 20 | ps | |
| RefClk input RMS jitter, bandwidth between 500 kHz and 15 MHz | | | | 4 | ps | |
| RefClk input RMS jitter, bandwidth between 15 MHz and 40 MHz | | | | 20 | ps | |

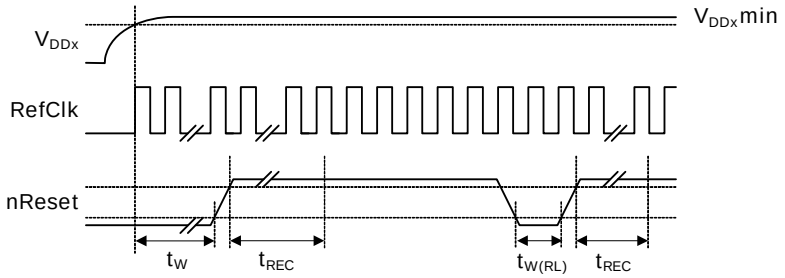
*Table 823 •* **Reference Clock AC Specifications *(continued)***

| Parameter | Symbol | Minimum | Typical | Maximum | Unit | Condition |
|---|---|---|---|---|---|---|
| RefClk input RMS jitter, bandwidth between 40 MHz and 80 MHz | | | | 100 | ps | |
| Jitter gain from RefClk to SerDes output, bandwidth between 0 MHz and 0.1 MHz | | | | 0.3 | dB | |
| Jitter gain from RefClk to SerDes output, bandwidth between 0.1 MHz and 7 MHz | | | | 3 | dB | |
| Jitter gain from RefClk to SerDes output, bandwidth above 7 MHz | | | | 3 – 20 × log (*f*/7 MHz) | dB | |

## 8.2.2 Reset Timing

The nReset signal waveform and the required measurement points for the timing specification are shown in the following illustration.

*Figure 83 •* **nReset Signal Timing Specifications**




The signal applied to the nReset input must comply with the specifications listed in the following table at the reset pin of the device.

*Table 824 •* **nReset Timing Specifications**

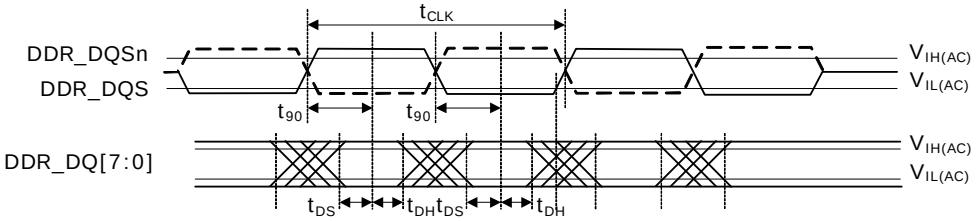
| Parameter | Symbol | Minimum | Maximum | Unit |
|---|---|---|---|---|
| nReset assertion time after power supplies and clock stabilize | $t_W$ | 2 | | ms |
| Recovery time from reset inactive to device fully active | $t_{REC}$ | | 50 | ms |
| nReset pulse width | $t_{W(RL)}$ | 100 | | ns |

## 8.2.3 DDR2 SDRAM Signal

This section provides the AC characteristics for the DDR2 SDRAM interface.

The following illustration shows the DDR2 SDRAM input timing diagram.

***Figure 84 •*** **DDR2 SDRAM Input Timing Diagram**



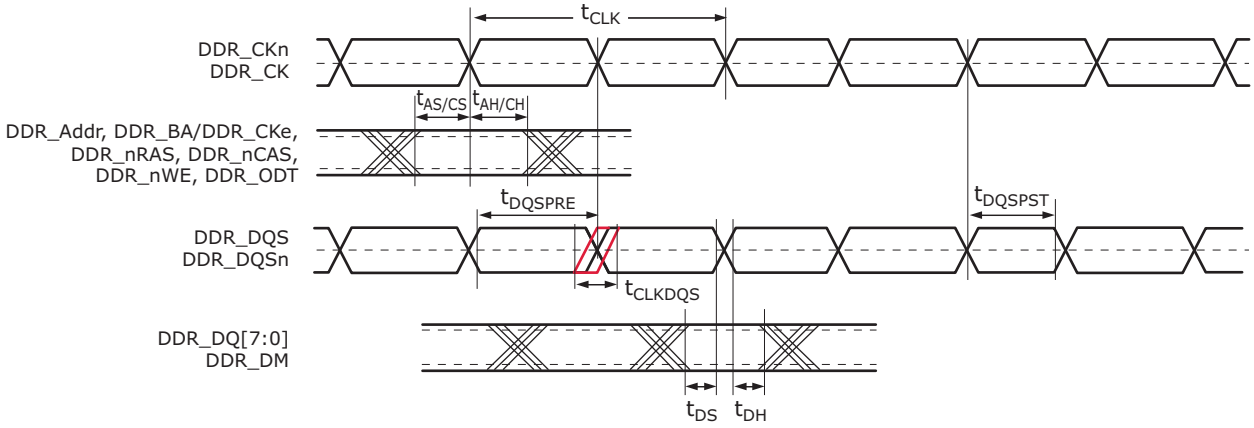

The following table lists the AC specifications for the DDR2 SDRAM input signals.

***Table 825 •*** **DDR2 SDRAM Input Signal AC Characteristics**

| Parameter | Symbol | Minimum | Maximum | Unit |
|---|---|---|---|---|
| Input voltage high | $V_{IH(AC)}$ | DDR_$V_{REF}$ + 0.20 | $V_{DD_IODDR}$ + 0.3 | V |
| Input voltage low | $V_{IL(AC)}$ | –0.3 | DDR_$V_{REF}$ – 0.20 | V |
| Differential input voltage | $V_{ID(AC)}$ | 0.5 | $V_{DD_IODDR}$ | V |
| Differential crosspoint voltage | $V_{IX(AC)}$ | 0.5 × $V_{DD_IODDR}$ – 0.175 | 0.5 × $V_{DD_IODDR}$ + 0.175 | V |
| DDR_DQ[7:0] input setup time relative to DDR_DQS/DDR_DQSn | $t_{DS}$ | | 350 | ps |
| DDR_DQ[7:0] input hold time relative to DDR_DQS/DDR_DQSn | $t_{DH}$ | | 250 | ps |

The following illustration shows the timing diagram for the DDR2 SDRAM outputs.

***Figure 85 •*** **DDR2 SDRAM Output Timing Diagram**




The following table lists the AC characteristics for the DDR2 SDRAM output signals.

***Table 826 •*** **DDR2 SDRAM Output Signal AC Characteristics**

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| DDR_CK cycle time 208 MHz (DDR400)(1) | $t_{CLK}$ | | 4.80 | | ns |
| DDR_CK/CKn duty cycle | | 48 | | 52 | % |

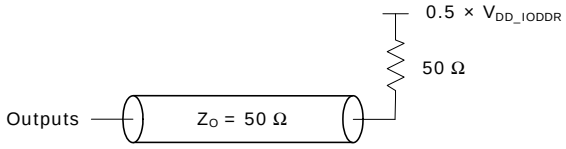
*Table 826 •* **DDR2 SDRAM Output Signal AC Characteristics *(continued)***

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| DDR_A, DDR_BA, DDR_CKe, DDR_nRAS, DDR_nCAS, DDR_ODT, and DDR_nWE output setup time relative to DDR_CK/CKn | $t_{AS}$ | 1000 | | | ps |
| DDR_A, DDR_BA, DDR_CKe, DDR_nRAS, DDR_nCAS, DDR_ODT, and DDR_nWE output hold time relative to DDR_CK/CKn | $t_{AH}$ | 1000 | | | ps |
| DDR_CK/CKn to DDR_DQS/DDR_DQSn skew | $t_{CLKDQS}$ | –600 | | 600 | ps |
| DDR_DQ[7:0]/DDR_DM output setup time with relative to DDR_DQS/DDR_DQSn | $t_{DS}$ | 700 | | | ps |
| DDR_DQ[7:0]/DDR_DM output hold time relative to DDR_DQS/DDR_DQSn | $t_{DH}$ | 700 | | | ps |
| DDR_DQS/DDR_DQSn preamble start | $t_{DQSPRE}$ | 0.4 × $t_{CLK}$ | | –0.6 × $t_{CLK}$ | ps |
| DDR_DQS/DDR_DQSn postamble end | $t_{DQSPST}$ | 0.4 × $t_{CLK}$ | | –0.6 × $t_{CLK}$ | ps |

1. Timing reference is DDR_CK/DDR_CKn crossing ±0.1 V.

The following illustration shows the test load circuit for the DDR2 outputs.

*Figure 86 •* **Test Load Circuit for DDR2 Outputs**

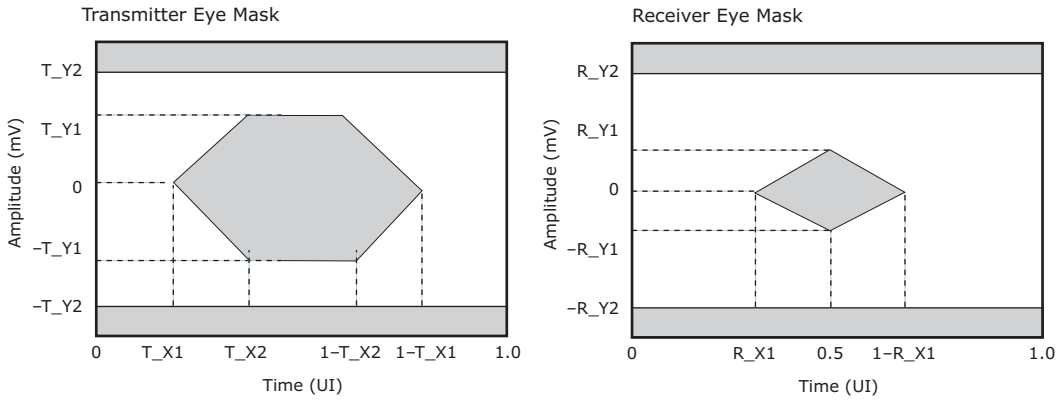



## 8.2.4 Enhanced SerDes Interface

All AC specifications for the Enhanced SerDes interface are compliant with the QSGMII Specification Revision 1.3 and meet or exceed the requirements in the standard. They are also compliant with the OIF-CEI version 2.0 requirements where applicable.

The Enhanced SerDes interface supports three major modes: SGMII, QSGMII,and SFP. The values in the tables in the following sections apply to modes listed in the condition column and are based on the test circuit shown in Figure 80, page 606. The transmit and receive eye specifications in the tables relate to the eye diagrams shown in the following illustration, with the compliance load as defined in the test circuit.

***Figure 87 •*** **QSGMII Transient Parameters**




### 8.2.4.1 Enhanced SerDes Outputs

The following table provides the AC specifications for the Enhanced SerDes outputs in SGMII mode.

***Table 827 •*** **Enhanced SerDes Output AC Specifications in SGMII Mode**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Unit interval, 1.25G | UI | | | | 800 ps. |
| $V_{OD}$ ringing compared to $V_S$ | $V_{RING}$ | | ±10 | % | $R_L$ = 100 Ω ±1%. |
| $V_{OD}$ rise time and fall time | $t_R$, $t_F$ | 100 | 200 | ps | 20% to 80% of $V_S$, $R_L$ = 100 Ω ±1%. |
| Differential output peak-to-peak voltage | $V_{OD}$ | | 30 | mV | Tx disabled. |
| Differential output return loss, 1000BASE-KX mode, 50 MHz to 625 MHz | $RL_{TX_DIFF}$ | ≥10 | | dB | $R_L$ = 100 Ω ±1%. |
| Differential output return loss, 1000BASE-KX mode, 625 MHz to 1250 MHz | $RL_{TX_DIFF}$ | 10–10 × log (f/625 MHz) | | dB | $R_L$ = 100 Ω ±1% |
| Common mode return loss, 1000BASE-KX mode | $RL_{CM}$ | 6 | | dB | 50 MHz to 625 MHz |
| Intrapair skew, SGMII mode | $t_{SKEW}$ | | 20 | ps | |

The following table provides the AC specifications for the Enhanced SerDes outputs in QSGMII mode.

***Table 828 •*** **Enhanced SerDes Output AC Specifications in QSGMII Mode**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Unit interval, 5G | UI | | | | 200 ps. |
| $V_{OD}$ rise time and fall time | $t_R$, $t_F$ | 30 | 96 | ps | 20% to 80% of $V_S$, $R_L$ = 100 Ω ±1%. |
| Differential output peak-to-peak voltage | $V_{OD}$ | | 30 | mV | Tx disabled. |
| Differential output return loss 100 MHz to 2.5 GHz | $RL_{TX_DIFF}$ | 8 | | dB | $R_L$ = 100 Ω ±1%. |

*Table 828 •* **Enhanced SerDes Output AC Specifications in QSGMII Mode**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Differential output return loss, 1000BASE-KX mode, 2.5 GHz to 5 GHz | $RL_{TX_DIFF}$ | 8 dB – 16.6 log (f/2.5 GHz) | | dB | $R_L$ = 100 Ω ±1%. |
| Eye mask (T_X1) | | | 0.15 | UI | |
| Eye mask (T_X2) | | | 0.4 | UI | |
| Eye mask (T_Y1) | | 200 | | mV | |
| Eye mask (T_Y2) | | | 450 | mV | |

### 8.2.4.2 Enhanced SerDes Driver Jitter Characteristics

The following table lists the jitter characteristics for the Enhanced SerDes driver in SGMII mode.

*Table 829 •* **Enhanced SerDes Driver Jitter Characteristics in SGMII Mode**

| Parameter | Symbol | Maximum | Unit | Condition |
|---|---|---|---|---|
| Total output jitter | $t_{JIT(O)}$ | 192 | ps | Measured according to IEEE 802.3.38.5. |
| Deterministic output jitter | $t_{JIT(OD)}$ | 80 | ps | Measured according to IEEE 802.3.38.5. |

The following table lists the jitter characteristics for the Enhanced SerDes driver in QSGMII mode.

*Table 830 •* **Enhanced SerDes Driver Jitter Characteristics in QSGMII Mode**

| Parameter | Symbol | Maximum | Unit | Condition |
|---|---|---|---|---|
| Total output jitter | $t_{JIT(O)}$ | 60 | ps | Measured according to IEEE 802.3.38.5. |
| Deterministic output jitter | $t_{JIT(OD)}$ | 10 | ps | Measured according to IEEE 802.3.38.5. |

### 8.2.4.3 Enhanced SerDes Inputs

The following table lists the AC specifications for the Enhanced SerDes inputs in SGMII mode.

*Table 831 •* **Enhanced SerDes Input AC Specifications in SGMII Mode**

| Parameter | Symbol | Minimum | Unit | Condition |
|---|---|---|---|---|
| Unit interval, 1.25G | UI | | ps | 800 ps. |
| Differential input return loss | $RL_{RX_DIFF}$ | 10 | dB | 50 MHz to 625 MHz, $R_L$ = 100 Ω ±1%. |
| Common-mode input return loss | | 6 | dB | 50 MHz to 625 MHz. |

The following table lists the AC specifications for the Enhanced SerDes inputs in QSGMII mode.

*Table 832 •* **Enhanced SerDes Input AC Specifications in QSGMII Mode**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Unit interval, 5G | UI | | | | 200 ps. |

*Table 832 •* **Enhanced SerDes Input AC Specifications in QSGMII Mode**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Differential input return loss, 100 MHz to 2.5 GHz | $RL_{RX_DIFF}$ | 8 | | dB | $R_L$ = 100 Ω ±1%. |
| Differential input return loss, 2.5 GHz to 5 GHz | $RL_{RX_DIFF}$ | 8 dB – 16.6 log (f/2.5 GHz) | | dB | $R_L$ = 100 Ω ±1%. |
| Common-mode input return loss | | 6 | | dB | 100 MHz to 2.5 GHz. |
| Eye mask (R_X1) | | | 0.3 | UI | |
| Eye mask (R_Y1) | | | 50 | mV | |
| Eye mask (R_Y2) | | | 450 | mV | |

### 8.2.4.4 Enhanced SerDes Receiver Jitter Tolerance

The following table lists jitter tolerances for the Enhanced SerDes receiver in SGMII mode.

*Table 833 •* **Enhanced SerDes Receiver Jitter Tolerance in SGMII Mode**

| Parameter | Symbol | Minimum | Unit | Condition |
|---|---|---|---|---|
| Total input jitter tolerance, 1000BASE-KX and SFP modes | $t_{JIT(I)}$ | 600 | ps | Above 637 kHz. Measured according to IEEE 802.3 38.6.8. |
| Deterministic input jitter tolerance, 1000BASE-KX and SFP mode | $t_{JIT(ID)}$ | 370 | ps | Above 637 kHz. Measured according to IEEE 802.3 38.6.8. |
| Cycle distortion input jitter tolerance, 100BASE-FX mode | $D_{CD}$ | 1.4 | ns | Measured according to ISO/IEC 9314-3:1990. IB_ENA_CMV_TERM = 1 IB_ENA_DC_COUPLING = 1 |
| Data-dependent input jitter tolerance, 100BASE-FX mode | $D_{DJ}$ | 2.2 | ns | Measured according to ISO/IEC 9314-3:1990. IB_ENA_CMV_TERM = 1 IB_ENA_DC_COUPLING = 1 |
| Random input jitter tolerance, peak-to-peak, 100BASE-FX mode | $R_J$ | 2.27 | ns | Measured according to. ISO/IEC 9314-3:1990. IB_ENA_CMV_TERM = 1 IB_ENA_DC_COUPLING = 1 |

The following table lists jitter tolerances for the Enhanced SerDes receiver in QSGMII mode.

*Table 834 •* **Enhanced SerDes Receiver Jitter Tolerance in QSGMII Mode**

| Parameter | Symbol | Maximum | Unit | Condition |
|---|---|---|---|---|
| Bounded high-probability jitter(1) | $BHP_J$ | 90 | ps | 92 ps peak-to-peak random jitter, and 38 ps sinusoidual jitter (SJHF). |
| Sinusoidal jitter, maximum | $SJ_{MAX}$ | 1000 | ps | |
| Sinusoidal jitter, high frequency | $SJ_{HF}$ | 10 | ps | |

*Table 834 •* **Enhanced SerDes Receiver Jitter Tolerance in QSGMII Mode**

| Parameter | Symbol | Maximum | Unit | Condition |
|---|---|---|---|---|
| Total input jitter tolerance | $t_{JIT(I)}$ | 120 | ps | 92 ps peak-to-peak random jitter, and 38 ps sinusoidal jitter (SJHF). |

1. This is the sum of uncorrelated bounded high probability jitter (0.15 UI) and correlated bounded high probability jitter (0.30 UI).
Uncorrelated bounded high probability jitter is defined as jitter distribution where the value of the jitter shows no correlation to any signal level being transmitted. Formally defined as deterministic jitter ($T_{_DJ}$).
Correlated bounded high probability jitter is defined as jitter distribution where the value of the jitter shows a strong correlation to the signal level being transmitted.

## 8.2.5 SerDes (SGMII) Interface

In SGMII mode, the SGMII interface is compliant with Serial-GMII Specification, version 1.9.

In 1000BASE-KX mode, the SGMII interface is compliant with IEEE 802.3 clause 70.

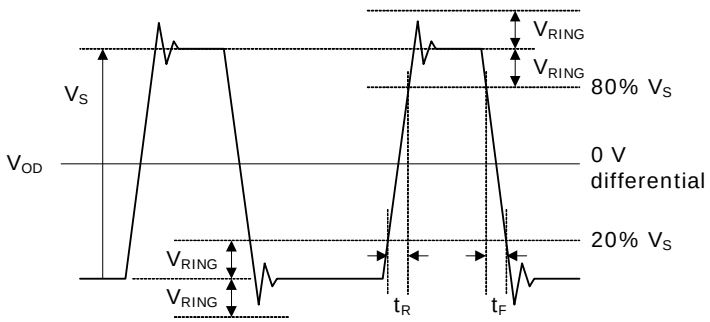
In SFP mode, the SGMII interface is compliant with the SFP MSA standard.

In 100BASE-FX mode, the SGMII interface is compliant with IEEE 802.3 clause 26.

The rise time and fall time parameters and other transient performance specifications are defined in the following illustration. The definition of $V_S$ is the difference between the steady state high and low voltage of the differential signal.

In addition, the signals are monotonic between 20% and 80% of $V_S$ when loaded with 100 Ω ±1%.

*Figure 88 •* **SGMII Transient Parameters**




All SerDes driver signals comply with the conditions listed in the following table when measured with the test circuit shown in Figure 80, page 606.

### 8.2.5.1 SerDes Outputs

The values in the following table are valid for all configurations, unless stated in the conditions column.

*Table 835 •* **SerDes Output AC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| $V_{OD}$ ringing compared to $V_S$, SGMII mode | $V_{RING}$ | | ±10 | % | $R_L$ = 100 Ω ±1%. |
| $V_{OD}$ rise time and fall time, SGMII mode | $t_R$, $t_F$ | 100 | 200 | ps | 20% to 80% of $V_S$, $R_L$ = 100 Ω ±1%. |
| Differential output peak-to-peak voltage | $V_{OD}$ | | 30 | mV | Tx disabled. |

*Table 835 •* **SerDes Output AC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Differential output return loss, 1000BASE-KX mode, 50 MHz to 625 MHz | $RL_{TX_DIFF}$ | ≥10 | | dB | $R_L$ = 100 Ω ±1%. |
| Differential output return loss, 1000BASE-KX mode, 625 MHz to 1250 MHz | $RL_{TX_DIFF}$ | 10–10 × log (*f*/625 MHz) | | dB | $R_L$ = 100 Ω ±1%. |
| Common-mode return loss, 1000BASE-KX mode | $RL_{CM}$ | 6 | | dB | 50 MHz to 625 MHz. |
| Intrapair skew, SGMII mode | $t_{SKEW}$ | | 20 | ps | |

### 8.2.5.2 SerDes Driver Jitter Characteristics

The following table lists the jitter characteristics for the SerDes driver.

*Table 836 •* **SerDes Driver Jitter Characteristics**

| Parameter | Symbol | Maximum | Unit | Condition |
|---|---|---|---|---|
| Total output jitter | $t_{JIT(O)}$ | 192 | ps | Measured according to IEEE 802.3.38.5. |
| Deterministic output jitter | $t_{JIT(OD)}$ | 80 | ps | Measured according to IEEE 802.3.38.5. |

### 8.2.5.3 SerDes Inputs

The following table lists the AC specifications for the SerDes inputs.

*Table 837 •* **SerDes Input AC Specifications**

| Parameter | Symbol | Maximum | Unit | Condition |
|---|---|---|---|---|
| Differential input return loss, 1000BASE-KX mode, 50 MHz to 625 MHz | | ≥10 | dB | $R_L$ = 100 Ω ±1%. |
| Differential input return loss, 1000BASE-KX mode, 625 MHz to 1250 MHz | | 10–10 × log (*f*/625 MHz) | dB | $R_L$ = 100 Ω ±1%. |

### 8.2.5.4 SerDes Receiver Jitter Tolerance

The following table lists jitter tolerances for the SerDes receiver.

*Table 838 •* **SerDes Receiver Jitter Tolerance**

| Parameter | Symbol | Minimum | Unit | Condition |
|---|---|---|---|---|
| Total input jitter tolerance, 1000BASE-KX and SFP modes | $t_{JIT(I)}$ | 600 | ps | Above 637 kHz. Measured according to IEEE 802.3 38.6.8. |
| Deterministic input jitter tolerance, 1000BASE-KX and SFP modes | $t_{JIT(ID)}$ | 370 | ps | Above 637 kHz. Measured according to IEEE 802.3 38.6.8. |

*Table 838 •* **SerDes Receiver Jitter Tolerance**

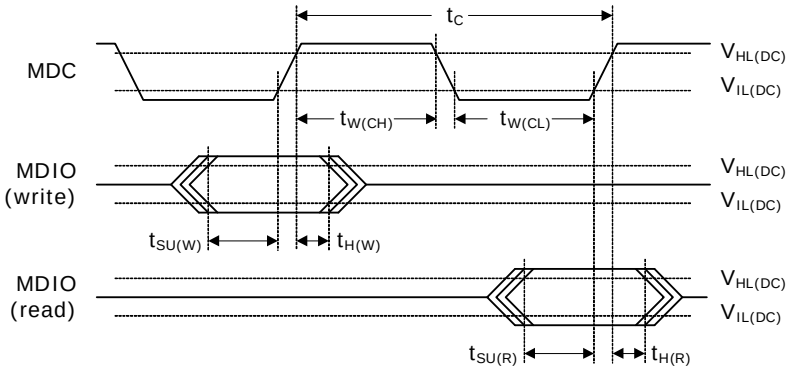
| Parameter | Symbol | Minimum | Unit | Condition |
|---|---|---|---|---|
| Cycle distortion input jitter tolerance, 100BASE-FX mode | $D_{CD}$ | 1.4 | ns | Measured according to ISO/IEC 9314-3:1990. |
| Data-dependent input jitter tolerance, 100BASE-FX mode | $D_{DJ}$ | 2.2 | ns | Measured according to ISO/IEC 9314-3:1990. |
| Random input jitter tolerance, peak-to-peak, 100BASE-FX mode | $R_J$ | 2.27 | ns | Measured according to ISO/IEC 9314-3:1990. |

## 8.2.6 MII Management

All AC specifications for the MII Management (MIIM) interface meet or exceed the requirements of IEEE 802.3-2002 (clause 22.2-4).

All MIIM AC timing requirements are specified relative to the input low and input high threshold levels. The following illustration shows the MIIM waveforms and required measurement points for the signals.

*Figure 89 •* **MIIM Timing Diagram**




The setup time of MDIO relative to the rising edge of MDC is defined as the length of time between when the MDIO exits and remains out of the switching region and when MDC enters the switching region. The hold time of MDIO relative to the rising edge of MDC is defined as the length of time between when MDC exits the switching region and when MDIO enters the switching region.

All MIIM signals comply with the specifications in the following table. The MDIO signal requirements are requested at the pin of the device.

*Table 839 •* **MIIM Timing Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| MDC frequency(1) | $f$ | 0.488 | 20.83 | MHz | |
| MDC cycle time(2) | $t_C$ | 48 | 2048 | ns | |
| MDC time high | $t_{W(CH)}$ | 20 | | ns | $C_L$ = 50 pF |
| MDC time low | $t_{W(CL)}$ | 20 | | ns | $C_L$ = 50 pF |
| MDC input rise and fall time for slave mode | $t_R$, $t_F$ | | 10 | ns | Between $V_{IL(MAX)}$ and $V_{IH(MIN)}$ |
| MDIO setup time to MDC on write | $t_{SU(W)}$ | 15 | | ns | $C_L$ = 50 pF |
| MDIO hold time from MDC on write | $t_{H(W)}$ | 15 | | ns | $C_L$ = 50 pF |

*Table 839 •* **MIIM Timing Specifications**

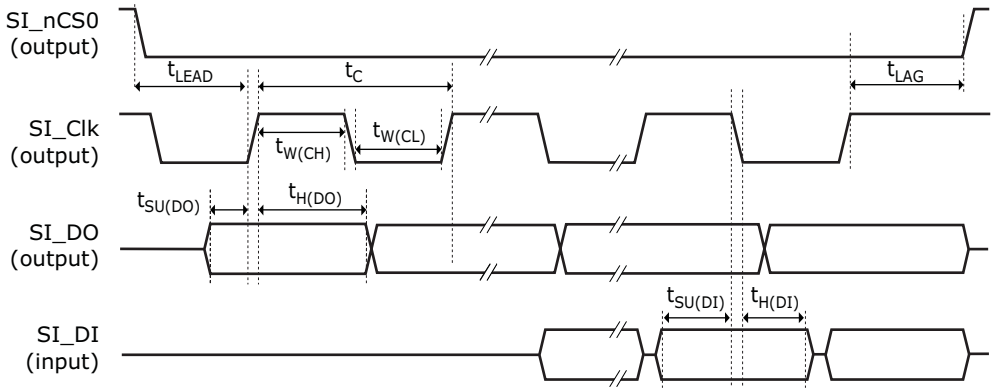
| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| MDIO setup time to MDC on read | $t_{SU(R)}$ | 30 | | ns | $C_L$ = 50 pF on MDC |
| MDIO hold time from MDC on read | $t_{H(R)}$ | 0 | | ns | $C_L$ = 50 pF |

1. For the maximum value, the devices support an MDC clock speed of up to 20 MHz for faster communication with the PHYs. If the standard frequency of 2.5 MHz is used, the MIIM interface is designed to meet or exceed the IEEE 802.3 requirements of the minimum MDC high and low times of 160 ns and an MDC cycle time of minimum 400 ns, which is not possible at faster speeds.
2. Calculated as $t_C = 1/f$.

## 8.2.7 Serial CPU Interface (SI) Master Mode

All serial CPU interface (SI) timing requirements for master mode are specified relative to the input low and input high threshold levels. The following illustration shows the timing parameters and measurement points.

*Figure 90 •* **SI Timing Diagram for Master Mode**




All SI signals comply with the specifications shown in the following table. The SI input timing requirements are requested at the pins of the device.

*Table 840 •* **SI Timing Specifications for Master Mode**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Clock frequency | $f$ | | 25(1) | MHz | |
| Clock cycle time | $t_C$ | 40 | | ns | |
| Clock time high | $t_{W(CH)}$ | 16 | | ns | |
| Clock time low | $t_{W(CL)}$ | 16 | | ns | |
| Clock rise time and fall time | $t_R$, $t_F$ | | 10 | ns | Between $V_{IL(MAX)}$ and $V_{IH(MIN)}$. $C_L$ = 30 pF. |
| DO setup time to clock | $t_{SU(DO)}$ | 10 | | ns | |
| DO hold time from clock | $t_{H(DO)}$ | 10 | | ns | |
| Enable active before first clock | $t_{LEAD}$ | 10 | | ns | |
| Enable inactive after clock | $t_{LAG}$ | 5 | | ns | |
| DI setup time to clock | $t_{SU(DI)}$ | 22 | | ns | |
| DI hold time from clock | $t_{H(DI)}$ | –2 | | ns | |

1. Frequency is programmable. The startup frequency is 4 MHz.

## 8.2.8 Serial CPU Interface (SI) for Slave Mode

All serial CPU interface (SI) slave mode timing requirements are specified relative to the input low and input high threshold levels. The following illustrations show the timing parameters and measurement points for SI input and output data.

***Figure 91 •*** **SI Input Data Timing Diagram for Slave Mode**

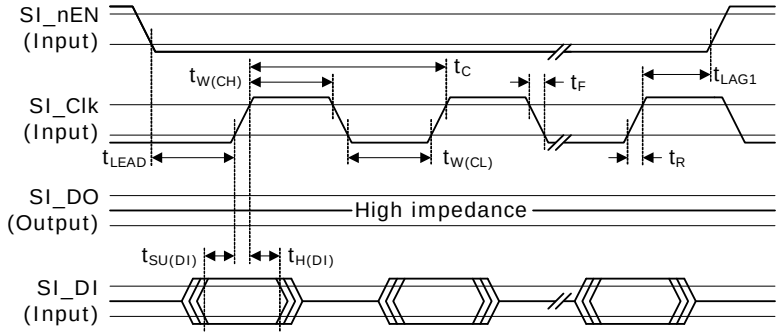
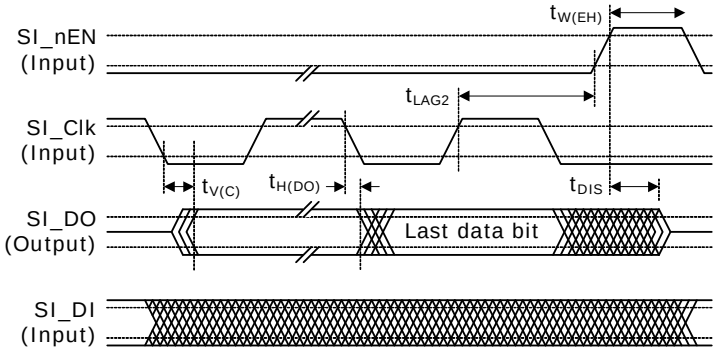



***Figure 92 •*** **SI Output Data Timing Diagram for Slave Mode**




All SI signals comply with the specifications shown in the following table. The SI input timing requirements are requested at the pins of the device.

***Table 841 •*** **SI Timing Specifications for Slave Mode**

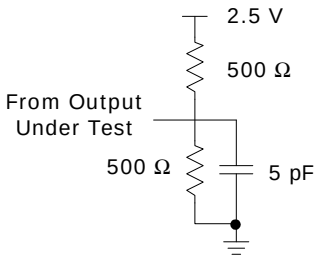
| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Clock frequency | $f$ | | 25 | MHz | |
| Clock cycle time | $t_C$ | 40 | | ns | |
| Clock time high | $t_{W(CH)}$ | 16 | | ns | |
| Clock time low | $t_{W(CL)}$ | 16 | | ns | |
| Clock rise time and fall time | $t_R$, $t_F$ | | 10 | ns | Between $V_{IL(MAX)}$ and $V_{IH(MIN)}$. |
| DI setup time to clock | $t_{SU(DI)}$ | 4 | | ns | |
| DI hold time from clock | $t_{H(DI)}$ | 4 | | ns | |
| Enable active before first clock | $t_{LEAD}$ | 10 | | ns | |
| Enable inactive after clock (input cycle)(1) | $t_{LAG1}$ | 25 | | ns | |
| Enable inactive after clock (output cycle) | $t_{LAG2}$ | See note(2) | | ns | |

*Table 841 •* **SI Timing Specifications for Slave Mode *(continued)***

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Enable inactive width | $t_{W(EH)}$ | 20 | | ns | |
| DO valid after clock | $t_{V(C)}$ | | 20 | ns | $C_L$ = 30 pF. |
| DO hold time from clock | $t_{H(DO)}$ | 0 | | ns | $C_L$ = 0 pF. |
| DO disable time(3) | $t_{DIS}$ | | 15 | ns | See Figure 93, page 624. |

1. $t_{LAG1}$ is defined only for write operations to the device, not for read operations.
2. The last rising edge on the clock is necessary for the external master to read in the data. The lag time depends on the necessary hold time on the external master data input.
3. Pin begins to float when a 300 mV change from the loaded $V_{OH}$ or $V_{OL}$ level occurs.

*Figure 93 •* **SI_DO Disable Test Circuit**




## 8.2.9 Parallel Interface (PI) Master Mode

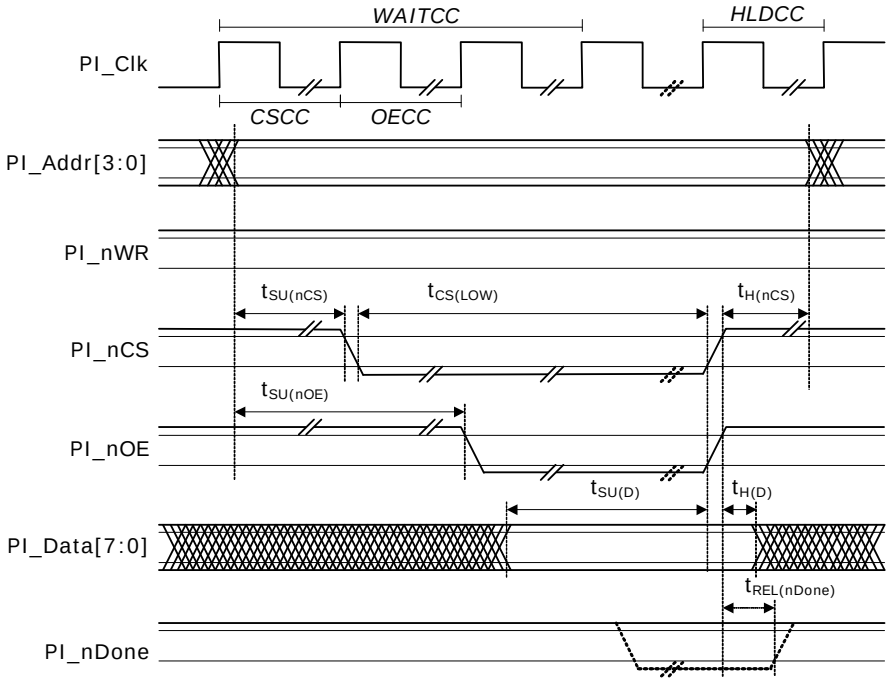
This section provides the AC timing specifications for the PI master mode signals: PI_nCS, PI_nWR, PI_nOE, PI_nDone, PI_Addr[3:0], and PI_Data[7:0]. The PI signals are alternate function signals on GPIO_[13-28] pins. For more information about the GPIO pin mapping, see the Pins by Function section for the appropriate device.

The timing specifications for parallel interface refer to the VCore-III CPU's external RAM/ROM interface. The timing is programmable and shown as defined by default register values.

### 8.2.9.1 VCore-III CPU External PI Read Access

The VCore-III CPU timing parameters and required measurement points for external PI read access are defined in the following illustration. All VCore-II CPU signals for external PI read accesses comply with the specifications in the table following the illustration.

***Figure 94 •*** **VCore-III CPU External PI Read Access Timing Diagram**




The timing related to VCore-III external PI access is programmable. The programmable delays adjust timing in steps of the PI_Clk period. The PI_Clk period is determined by the dividers in the HSIO::PLL5G_CFG0 and ICPU_CFG::PI_MST_CFG registers. The default settings correspond to a PI_Clk period of 297.6 ns. The condition used for these specifications corresponds to a PI_Clk period of 22.4 ns. Additionally, the default delay settings are used for WAITCC(1), CSCC(1), OECC(0) and HLDCC(0) as defined by the PI_MST_CTRL registers.

***Table 842 •*** **VCore-III CPU External PI Read Timing Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Address/control setup time to chip select(1) | $t_{SU(nCS)}$ | 18 | | ns | $C_L$ = 30 pF |
| Address/control hold time from chip select(2) | $t_{H(nCS)}$ | –4 | | ns | $C_L$ = 30 pF |
| Address/control setup time to output enable(3) | $t_{SU(nOE)}$ | 18 | | ns | $C_L$ = 30 pF |
| Address/control hold time from output enable(4) | $t_{H(nOE)}$ | –4 | | ns | $C_L$ = 30 pF |
| Chip select low(5) | $t_{CS(low)}$ | 18 | 23 | ns | $C_L$ = 30 pF |
| Data setup time to chip select high | $t_{SU(D)}$ | 25 | | ns | $C_L$ = 30 pF |
| Data hold time from chip select high | $t_{h(D)}$ | 0 | | ns | $C_L$ = 30 pF |
| PI_nDone release after chip select high(6) | $t_{REL(nDone)}$ | 0 | | ns | $C_L$ = 30 pF |

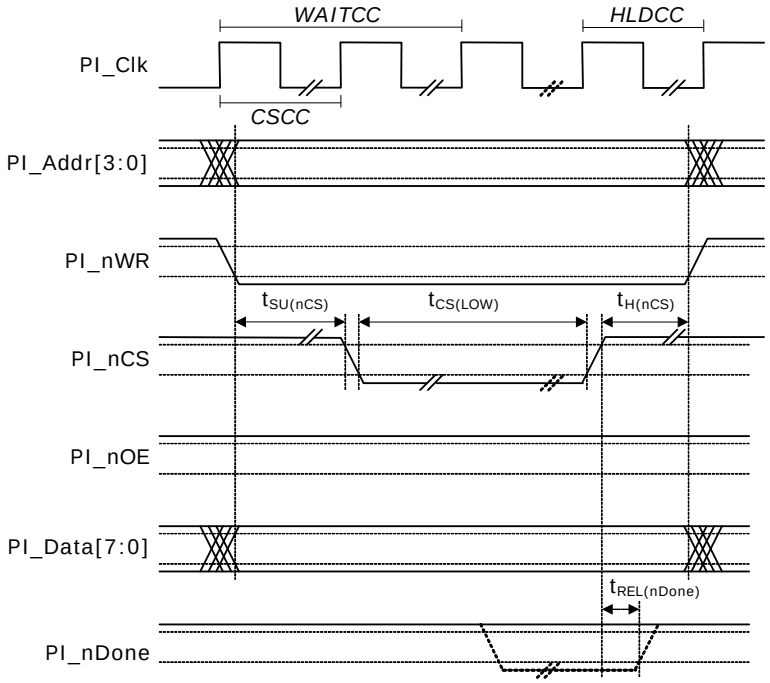
1. The minimum setup time of PI_Addr[3:0]/PI_nBE[1:0]/PI_nWR to PI_nCS low may be expressed as WAITCC × 22.4 ns – 4 ns = 18.4 ns.
2. The minimum hold time of PI_Addr[3:0]/PI_nBE[1:0]/PI_nWR from PI_nCS high may be expressed as HLDCC × 22.4 ns – 4 ns = –4 ns.
3. The minimum setup time of PI_Addr[3:0]/PI_nBE[1:0]/PI_nWR to PI_nOE low may be expressed as (WAITCC + OECC) × 22.4 ns – 4 ns = 18.4 ns.

4. The minimum hold time of PI_ADDR[3:0]/PI_nBE[1:0]/PI_nWR from PI_nOE high may be expressed as HLDCC × 22.4 ns – 4 ns = –4 ns.
5. The maximum PI_nCS low time may expressed as (WAITCC + 1 – CSCC) × 22.4 ns = 22.4 ns. The minimum is maximum 4 ns less than the maximum.
6. The interface can operate in a device-paced mode according to the PI_MST_CTRL registers. Device-paced mode allows slow devices to delay the access cycle termination beyond the WAITCC setting. A timeout can be specified in the PI_MST_CTRL registers to terminate access cycles from non-responsive external devices. In device-paced mode, PI_nDone must be released after PI_nCS is observed high and before the next access cycle is started. Slow devices may require HLDCC to be adjusted accordingly.

#### 8.2.9.2 VCore-III CPU External PI Write Access

The VCore-III CPU timing parameters and required measurement points for external PI write access are defined in the following illustration. All VCore-III CPU signals for the external PI write access comply with the specifications in the following table following the illustration.

***Figure 95 •*** **VCore-III CPU ROM/Flash Write Timing Diagram**




The timing related to VCore-III external PI access is programmable. The programmable delays adjust timing in steps of the PI_Clk period. The PI_Clk period is determined by the dividers in the HSIO::PLL5G_CFG0 and ICPU_CFG::PI_MST_CFG registers. The default settings correspond to a PI_Clk period of 297.6 ns. The condition used for these specifications corresponds to a PI_Clk period of 22.4 ns. Additionally, the default delay settings are used for WAITCC(1), CSCC(1), OECC(0) and HLDCC(0) as defined by the PI_MST_CTRL registers.

***Table 843 •*** **VCore-III CPU External PI Write Timing Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Address/control setup time to chip select(1) | $t_{SU(nCS)}$ | 18 | | ns | $C_L$ = 30 pF |
| Address/control hold time from chip select(2) | $t_{H(nCS)}$ | –4 | | ns | $C_L$ = 30 pF |
| Chip select low(3) | $t_{CS(low)}$ | 18 | 23 | ns | $C_L$ = 30 pF |
| Data setup time to chip select high | $t_{SU(D)}$ | 15 | | ns | $C_L$ = 30 pF |

*Table 843 •* **VCore-III CPU External PI Write Timing Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| PI_nDone release after chip select high(4) | $t_{REL(nDone)}$ | 0 | | ns | $C_L$ = 30 pF |

1. The minimum setup time of PI_ADDR[3:0]/PI_nBE[1:0]/PI_nWR to PI_nCSlow may be expressed as WAITCC × 22.4 ns – 4 ns = 18.4 ns.
2. The minimum hold time of PI_ADDR[3:0]/PI_nBE[1:0]/PI_nWR from PI_nCS high may be expressed as HLDCC × 22.4 ns – 4 ns = –4 ns.
3. The maximum PI_nCS low time may expressed as (WAITCC + 1 – CSCC) × 22.4 ns = 22.4 ns. The minimum is maximum 4 ns less than the maximum.
4. The interface can operate in a device-paced mode according to the PI_MST_CTRL registers. Device-paced mode allows slow devices to delay the access cycle termination beyond the WAITCC setting. A timeout can be specified in the PI_MST_CTRL registers to terminate access cycles from non-responsive external devices.In device-paced mode, PI_nDone must be released after PI_nCS is observed high and before the next access cycle is started. Slow devices may require HLDCC to be adjusted accordingly.

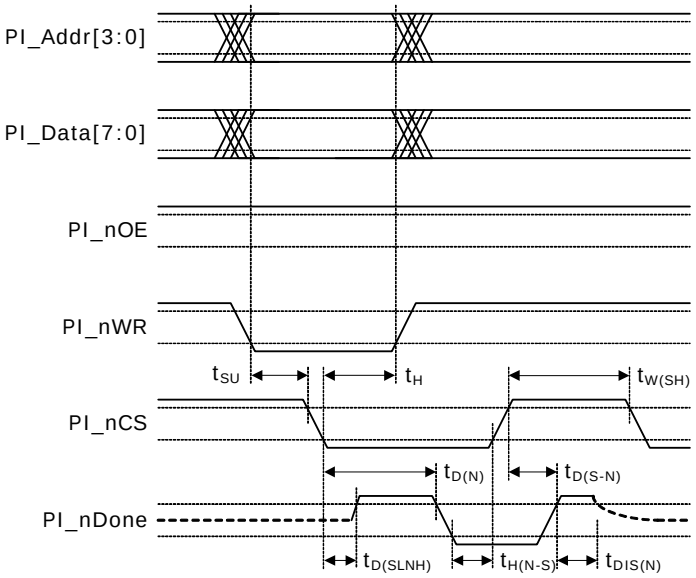
## 8.2.10 Parallel Interface (PI) Slave Mode

This section provides the AC timing specifications for the PI slave mode signals: PI_nCS, PI_nWR, PI_nOE, PI_nDone, PI_Addr[3:0], and PI_Data[7:0]. The PI signals are alternate function signals on the GPIO_[13:28] pins. For more information about the GPIO pin mapping, see the Pins by Function section for the appropriate device.

The AC timing specifications apply when an external CPU accesses the parallel CPU interface (slave mode operation).

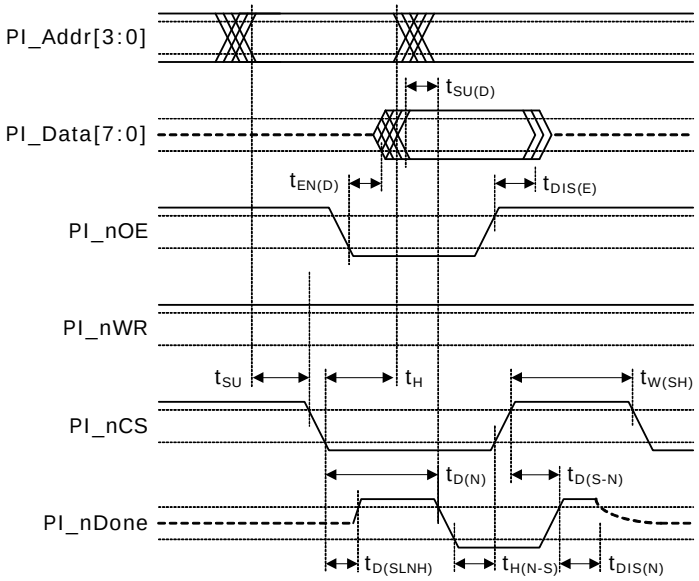
All PI timing specifications are relative to the input low and input high threshold levels. The following two illustrations show the PI timing parameters and the required measurement points.

*Figure 96 •* **PI Slave Write Cycle Timing Diagram**




PI_nDone's polarity is programmable, shown here as active low.

***Figure 97 •*** **PI Slave Read Cycle Timing Diagram**




PI_nDone's polarity is programmable, shown here as active low.

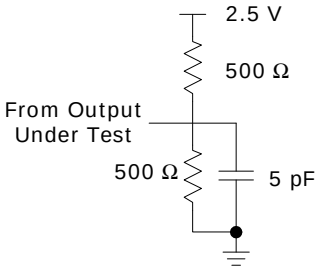
All PI signals comply with the timing parameters specified in the following table. The PI receive signal requirements are requested at the pin of the device.

***Table 844 •*** **PI Slave Mode Timing Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| PI_ADDR, PI_DATA, and PI_nWR setup to PI_nCS falling(1) | $t_{SU}$ | 4 | | ns | Data only on write. |
| PI_ADDR, PI_DATA, and PI_nWR hold from nCS low(1) | $t_{H}$ | 25 | | ns | Data only on write. |
| Delay from PI_nCS low to PI_nDone rising(2) | $t_{D(SLNH)}$ | | 25 | ns | $C_L$ = 30 pF. |
| Delay from PI_nCS low to PI_nDone falling(2) | $t_{D(N)}$ | | 55 | ns | $C_L$ = 30 pF. |
| PI_nCS hold from PI_nDone falling(1), (2), (3) | $t_{H(N-S)}$ | 0 | | ns | |
| Delay from PI_nCS high to PI_nDone high(2) | $t_{D(S-N)}$ | | 25 | ns | $C_L$ = 30 pF. |
| PI_nDone disable time from PI_nDone pulled inactive(2), (4) | $t_{DIS(N)}$ | | 12 | ns | See Figure 98, page 629. |
| Width of nCS high | $t_{W(SH)}$ | 10 | | ns | |
| PI_nOE and PI_nCS low to data enabled(1), (5) | $t_{EN(D)}$ | | 20 | ns | $C_L$ = 30 pF. |
| Data setup time to PI_nDone falling on read(2) | $t_{SU(D)}$ | 0 | | ns | $C_L$ = 30 pF. |
| Data disable time from either PI_nCS high or PI_nOE high(5) | $t_{DIS(E)}$ | | 20 | ns | See Figure 98, page 629. |

1. Before input data or conditions are sampled, an initial delay can be added in steps of 8 ns from 0 ns to 120 ns. The default delay is 104 ns to ensure operation with slow CPUs. Timing values in this table are shown with 0 ns delay.
2. PI_nDone polarity is programmable; it is shown as active low in the timing diagrams.
3. When using extended bus cycles, the response time can be up to 470 ns.
4. Pin begins to float when a 300 mV change from the loaded $V_{OH}/V_{OL}$ level occurs.
5. Internal data output enable requires both nCS and nOE active. A time of 15 ns is valid only if PI_WAIT in the PI_CFG register. If set to a value other than 0×00, the value shown for $t_{EN(D)}$ changes.

***Figure 98 •*** **Signal Disable Test Circuit**

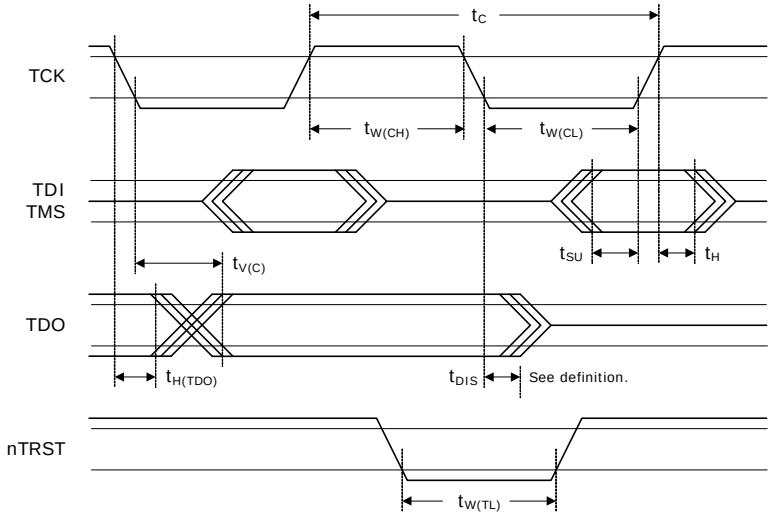



## 8.2.11 JTAG Interface

All AC specifications for the JTAG interface meet or exceed the requirements of IEEE 1149.1-2001.

The following illustration shows the JTAG transmit and receive waveforms and required measurement points for the different signals.

***Figure 99 •*** **JTAG Interface Timing Diagram**




All JTAG signals comply with the specifications in the following table. The JTAG receive signal requirements are requested at the pin of the device.

The JTAG_nTRST signal is asynchronous to the clock and does not have a setup or hold time requirement.

***Table 845 •*** **JTAG Interface AC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| TCK frequency | $f$ | | 10 | MHz | |
| TCK cycle time | $t_C$ | 100 | | ns | |
| TCK high time | $t_{W(CH)}$ | 40 | | ns | |

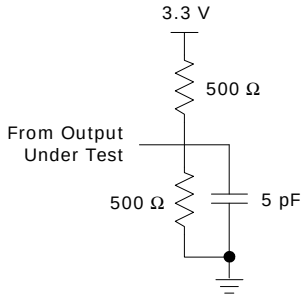
*Table 845 •* **JTAG Interface AC Specifications *(continued)***

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| TCK low time | $t_{W(CL)}$ | 40 | | ns | |
| Setup time to TCK rising | $t_{SU}$ | 10 | | ns | |
| Hold time from TCK rising | $t_H$ | 10 | | ns | |
| TDO valid after TCK falling | $t_{V(C)}$ | | 28 | ns | $C_L$ = 10 pF |
| TDO hold time from TCK falling | $t_{H(TDO)}$ | 0 | | ns | $C_L$ = 0 pF |
| TDO disable time(1) | $t_{DIS}$ | | 30 | ns | See Figure 100, page 630. |
| nTRST time low | $t_{W(TL)}$ | 30 | | ns | |

1. The pin begins to float when a 300 mV change from the actual $V_{OH}/V_{OL}$ level occurs.

The following illustration shows the test circuit for the TDO disable time.

*Figure 100 •* **Test Circuit for TDO Disable Time**

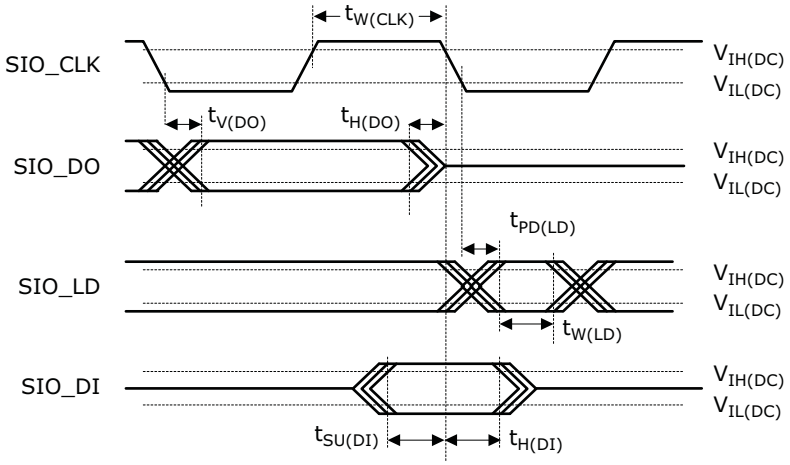



## 8.2.12 Serial Inputs/Outputs

This section provides the AC characteristics for the serial I/O signals: SIO_CLK, SIO_LD, SIO_DO, and SIO_DI. The SI signals are alternate function signals on the GPIO_[0:3] pins. For more information about the GPIO pin mapping, see the Pins by Function section for the appropriate device.

The serial I/O timing diagram is shown in the following illustration.

*Figure 101 •* **Serial I/O Timing Diagram**



The following table lists the serial I/O timing specifications.

***Table 846 •*** **Serial I/O Timing Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Clock frequency(1) | $f$ | | 25 | MHz | |
| SIO_CLK clock pulse width | $t_{W(CLK)}$ | 16 | | ns | 25 MHz clock |
| SIO_DO valid after clock falling | $t_{V(DO)}$ | | 6 | ns | |
| SIO_DO hold time from clock falling | $t_{H(DO)}$ | | 6 | ns | |
| SIO_LD propagation delay from clock falling | $t_{PD(LD)}$ | 40 | | ns | |
| SIO_LD width | $t_{W(LD)}$ | 10 | | ns | |
| SIO_DI setup time to clock | $t_{SU(DI)}$ | 25 | | ns | |
| SIO_DI hold time from clock | $t_{H(DI)}$ | 4 | | ns | |

1. The SIO clock frequency is programmable.

## 8.2.13 Two-Wire Serial Interface

This section provides the AC specifications for the two-wire serial interface signals TWI_SCL and TWI_SDA. The two-wire serial interface signals are alternate function signals on the GPIO_5 and GPIO_6 pins. For more information about the GPIO pin mapping, see the Pins by Function section for the appropriate device.

The two-wire serial interface signals are compatible with the Philips $I^2C$-BUS specifications, except for the minimum rise time and fall time requirements for fast mode.

***Figure 102 •*** **Two-Wire Serial Read Timing Diagram**

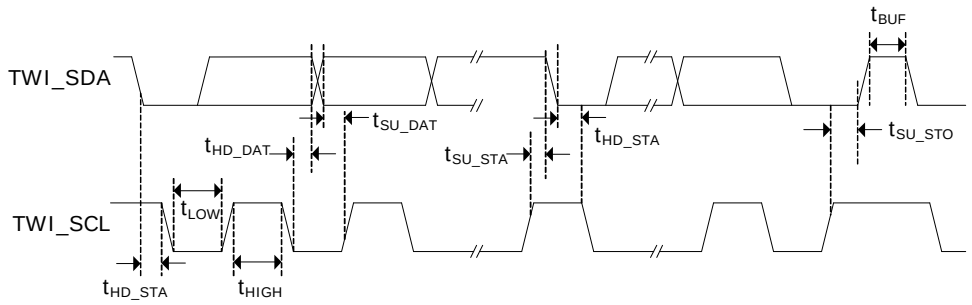
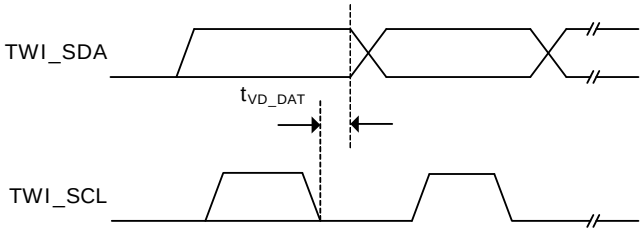



***Figure 103 •*** **Two-Wire Serial Write Timing Diagram**




For the specifications listed in the following table, standard mode is defined as 100 kHz and fast mode is 400 kHz. The data in this table assumes that the software-configurable two-wire interface timing parameters, SS_SCL_HCNT, SS_SCL_LCNT, FS_SCL_HCNT, and FS_SCL_LCNT, are set to valid

values for the selected speed. For more information about setting the values for the selected speed, see Table 693, page 523 through Table 696, page 524.

***Table 847 •*** **Two-Wire Serial Interface AC Specifications**

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| TWI_SCL clock frequency, standard mode | *f* | | 100 | kHz | |
| TWI_SCL clock frequency, fast mode | *f* | | 400 | kHz | |
| TWI_SCL low period, standard mode | $t_{LOW}$ | 4.7 | | µs | |
| TWI_SCL low period, fast mode | $t_{LOW}$ | 1.3 | | µs | |
| TWI_SCL high period, standard mode | $t_{HIGH}$ | 4.0 | | µs | |
| TWI_SCL high period, fast mode | $t_{HIGH}$ | 0.6 | | µs | |
| TWI_SCL and TWI_SDA rise time, standard mode | | | 1000 | ns | |
| TWI_SCL and TWI_SDA rise time, fast mode | | | 300 | ns | |
| TWI_SCL and TWI_SDA fall time, standard mode | | | 300 | ns | |
| TWI_SDA setup time to TWI_SCL fall, standard mode | $t_{SU_DAT}$ | 250 | | ns | |
| TWI_SDA setup time to TWI_SCL fall, fast mode | $t_{SU_DAT}$ | 100 | 300 | ns | |
| TWI_SDA hold time to TWI_SCL fall, standard mode(1) | $t_{HD_DAT}$ | 300 | 3450 | ns | 300 ns delay enabled in ICPU_CFG::TWI_CONFIG register. |
| TWI_SDA hold time to TWI_SCL fall, fast mode(1) | $t_{HD_DAT}$ | 300 | 900 | ns | 300 ns delay enabled in ICPU_CFG::TWI_CONFIG register. |
| Setup time for repeated START condition, standard mode | $t_{SU_STA}$ | 4.7 | | µs | |
| Setup time for repeated START condition, fast mode | $t_{SU_SAT}$ | 0.6 | | µs | |
| Hold time after repeated START condition, standard mode | $t_{HD_STA}$ | 4.0 | | µs | |
| Hold time after repeated START condition, fast mode | $t_{HD_STA}$ | 0.6 | | µs | |
| Bus free time between STOP and START conditions, standard mode | $t_{BUF}$ | 4.7 | | µs | |

*Table 847 •* **Two-Wire Serial Interface AC Specifications *(continued)***

| Parameter | Symbol | Minimum | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| Bus free time between STOP and START conditions, fast mode | $t_{BUF}$ | 1.3 | | µs | |
| Clock to valid data out, standard and fast modes(2) | $t_{VD_DAT}$ | 300 | | ns | |
| Pulse width of spike suppressed by input filter on TWI_SCL or TWI_SDA | | 0 | 5 | ns | |

1. An external device must provide a hold time of at least 300 ns for the TWI_SDA signal to bridge the undefined region of the falling edge of the TWI_SCL signal.
2. Some external devices may require more data in hold time (target device's $t_{HD_DAT}$) than what is provided by $t_{VD_DAT}$, for example, 300 ns to 900 ns. The minimum value of $t_{VD_DAT}$ is adjustable; the typical value given represents the recommended minimum value, which is enabled in CPU_CFG::TWI_CONFIG.

# 8.3 Current and Power Consumption

This section provides the current and power consumption requirements for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

## 8.3.1 Current Consumption

This section provides the operating current consumption parameters for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

Typical current consumption values are over nominal supply settings at 25 °C case temperature, and maximum traffic load. Maximum current consumption values are over worst-case process, temperature, and supply settings, and maximum traffic load.

The following table lists the typical and maximum operating current consumption values for the VC7424-02 device.

*Table 848 •* **Operating Current for VSC7424-02**

| Parameter | Symbol | Typical | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| $V_{DD}$ operating current | $I_{DD}$ | 1.3 | 2.1 | A | $V_{TYP}$ = 1.0 V |
| $V_{DD_A}$ operating current | $I_{DD_A}$ | 0.16 | 0.27 | A | $V_{TYP}$ = 1.0 V |
| $V_{DD_AL}$ operating current | $I_{DD_AL}$ | 0.16 | 0.25 | A | $V_{TYP}$ = 1.0 V |
| $V_{DD_AH}$ operating current | $I_{DD_AH}$ | 0.9 | 0.9 | A | $V_{TYP}$ = 2.5 V |
| $V_{DD_VS}$ operating current | $I_{DD_VS}$ | 0.13 | 0.13 | A | $V_{TYP}$ = 1.0 V or 1.2 V |
| $V_{DD_IODDR}$ operating current(1) | $I_{DD_IODDR}$ | 0.1 | 0.1 | A | $V_{TYP}$ = 1.8 V |
| $V_{DD_IO}$ operating current | $I_{DD_IO}$ | 0.1 | 0.1 | A | $V_{TYP}$ = 2.5 V |

1. DDR2 on-die termination is disabled.

The following table lists the typical and maximum operating current consumption values for the VC7425-02, VSC7426-02, and VSC7427-02 devices.

*Table 849 •* **Operating Current for VSC7425-02, VSC7426-02, and VSC7427-02**

| Parameter | Symbol | Typical | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| $V_{DD}$ operating current | $I_{DD}$ | 1.8 | 2.7 | A | $V_{TYP}$ = 1.0 V |
| $V_{DD_A}$ operating current | $I_{DD_A}$ | 0.22 | 0.27 | A | $V_{TYP}$ = 1.0 V |

*Table 849 •* **Operating Current for VSC7425-02, VSC7426-02, and VSC7427-02 *(continued)***

| Parameter | Symbol | Typical | Maximum | Unit | Condition |
|---|---|---|---|---|---|
| $V_{DD_AL}$ operating current | $I_{DD_AL}$ | 0.2 | 0.25 | A | $V_{TYP}$ = 1.0 V |
| $V_{DD_AH}$ operating current | $I_{DD_AH}$ | 1.4 | 1.6 | A | $V_{TYP}$ = 2.5 V |
| $V_{DD_VS}$ operating current | $I_{DD_VS}$ | 0.15 | 0.15 | A | $V_{TYP}$ = 1.0 V or 1.2 V |
| $V_{DD_IODDR}$ operating current(1) | $I_{DD_IODDR}$ | 0.1 | 0.1 | A | $V_{TYP}$ = 1.8 V |
| $V_{DD_IO}$ operating current | $I_{DD_IO}$ | 0.1 | 0.1 | A | $V_{TYP}$ = 2.5 V |

1. DDR2 on-die termination is disabled.

## 8.3.2 Power Consumption

This section provides the power consumption parameters for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices, based on current consumption and with DDR2 on-die termination disabled.

Typical power consumption values are over nominal supplies and 25 °C case temperature. Maximum power consumption values are over maximum temperature and all supplies at maximum voltages.

The following table lists the typical and maximum power consumption values for the VC7424-02 device.

*Table 850 •* **Power Consumption for VSC7424-02**

| Parameter | Typical | Maximum | Unit |
|---|---|---|---|
| Power consumption, SGMII in LVDS mode $V_{DD_VS}$ = 1.0 V | 4.4 | 5.8 | W |
| Power consumption, SGMII in high-drive mode $V_{DD_VS}$ = 1.2 V | 4.5 | 5.9 | W |

The following table lists the typical and maximum power consumption values for the VC7425-02, VSC7426-02, and VSC7427-02 devices.

*Table 851 •* **Power Consumption for VSC7425-02, VSC7426-02, and VSC7427-02**

| Parameter | Typical | Maximum | Unit |
|---|---|---|---|
| Power consumption, SGMII in LVDS mode $V_{DD_VS}$ = 1.0 V | 6.3 | 8.4 | W |
| Power consumption, SGMII in high-drive mode $V_{DD_VS}$ = 1.2 V | 6.4 | 8.5 | W |

## 8.3.3 Power Supply Sequencing

During power on and off, $V_{DD_A}$ and $V_{DD_VS}$ must never be more than 300 mV above $V_{DD}$.

$V_{DD_VS}$ must be powered, even if the associated interface is not used. These power supplies must not remain at ground or left floating.

A maximum delay of 100 ms from $V_{DD_IODDR}$ to $V_{DD}$ is recommended. There is no requirement from $V_{DD}$ to $V_{DD_IODDR}$.

There are no sequencing requirements for $V_{DD_AL}$, $V_{DD_AH}$, and $V_{DD_IO}$. These power supplies can remain at ground or left floating if not used.

The nReset and JTAG_nTRST inputs must be held low until all power supply voltages have reached their recommended operating condition values.

## 8.4 Operating Conditions

The following table lists the recommended operating conditions.

***Table 852 •*** **Recommended Operating Conditions**

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| Power supply voltage for core supply | $V_{DD}$ | 0.95 | 1.00 | 1.05 | V |
| Power supply voltage for analog circuits | $V_{DD_A}$ | 0.95 | 1.00 | 1.05 | V |
| Power supply voltage for analog circuits in twisted pair interface | $V_{DD_AL}$ | 0.95 | 1.00 | 1.05 | V |
| Power supply voltage for analog driver in twisted pair interface | $V_{DD_AH}$ | 2.38 | 2.50 | 2.62 | V |
| Power supply voltage for SerDes and Enhanced SerDes interfaces, 1.0 V(1) | $V_{DD_VS}$ | 0.95 | 1.00 | 1.05 | V |
| Power supply voltage for SerDes and Enhanced SerDes interfaces, 1.2 V | $V_{DD_VS}$ | 1.14 | 1.20 | 1.26 | V |
| Power supply voltage for DDR2 interface | $V_{DD_IODDR}$ | 1.70 | 1.80 | 1.90 | V |
| Power supply voltage for MIIM, PI, and miscellaneous I/O | $V_{DD_IO}$ | 2.38 | 2.50 | 2.62 | V |
| Operating temperature(2) | T | 0 | | 125 | °C |

1. The 1.0 V power supply for the enhanced SerDes interface is enabled in HSIO::SERDES6G_OB_CFG.OB_ENA1V_MODE.
2. Minimum specification is ambient temperature, and the maximum is junction temperature.

## 8.5 Stress Ratings

**Warning** Stresses listed in the following table may be applied to devices one at a time without causing permanent damage. Functionality at or exceeding the values listed is not implied. Exposure to these values for extended periods may affect device reliability.

***Table 853 •*** **Stress Ratings**

| Parameter | Symbol | Minimum | Maximum | Unit |
|---|---|---|---|---|
| Power supply voltage for core supply | $V_{DD}$ | –0.3 | 1.10 | V |
| Power supply voltage for analog circuits | $V_{DD_A}$ | –0.3 | 1.10 | V |
| Power supply voltage for analog circuits in twisted pair interface | $V_{DD_AL}$ | –0.3 | 1.10 | V |
| Power supply voltage for analog circuits in twisted pair interface | $V_{DD_AH}$ | –0.3 | 2.75 | V |
| Power supply voltage for SerDes and Enhanced SerDes interfaces | $V_{DD_VS}$ | –0.3 | 1.32 | V |
| Power supply voltage for DDR2 interface | $V_{DD_IODDR}$ | –0.3 | 1.98 | V |
| Power supply voltage for MIIM, PI, and miscellaneous I/O | $V_{DD_IO}$ | –0.3 | 2.75 | V |
| Storage temperature | $T_S$ | –55 | 125 | °C |
| Electrostatic discharge voltage, charged device model | $V_{ESD_CDM}$ | –250 | 250 | V |

*Table 853 •* **Stress Ratings** ***(continued)***

| Parameter | Symbol | Minimum | Maximum | Unit |
|---|---|---|---|---|
| Electrostatic discharge voltage, human body model | $V_{ESD_HBM}$ | –1750 | 1750 | V |

**Warning** This device can be damaged by electrostatic discharge (ESD) voltage. Microsemi recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures may adversely affect reliability of the device.

# 9 Pin Descriptions for VSC7424-02

The VSC7424-02 device has 672 pins, which are described in this section.

The pin information is also provided as an attached Microsoft Excel file, so that you can copy it electronically. In Adobe Reader, double-click the attachment icon.

## 9.1 Pin Diagram for VSC7424-02

The following illustration shows the pin diagram for the VSC7424-02 device. For clarity, the device is shown in two halves, the top left and top right.

***Figure 104*** • **Pin Diagram for VSC7424-02, Top Left**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | Reserved_57 | Reserved_55 | Reserved_53 | Reserved_51 | P7_D0P | P7_D1P | P7_D2P | P7_D3P | P6_D0P | P6_D1P | P6_D2P | P6_D3P |
| B | VSS_1 | Reserved_56 | Reserved_54 | Reserved_52 | Reserved_50 | P7_D0N | P7_D1N | P7_D2N | P7_D3N | P6_D0N | P6_D1N | P6_D2N | P6_D3N |
| C | Reserved_59 | Reserved_58 | COMA_MODE | nRESET | VDD_IO_21 | Reserved_1 | VCORE_CFG0 | VCORE_CFG1 | VCORE_CFG2 | VCore_ICE_nEn | Reserved_4 | RefClk_Sel0 | RefClk_Sel1 |
| D | Reserved_61 | Reserved_60 | Reserved_205 | VDD_AH_1 | VDD_AH_2 | Reserved_206 | Reserved_207 | Reserved_208 | Reserved_209 | Reserved_248 | VDD_AH_4 | Reserved_211 | Reserved_13 |
| E | Reserved_63 | Reserved_62 | Reserved_216 | VDD_AH_7 | VDD_AH_8 | VDD_IO_1 | VDD_IO_2 | VDD_AH_9 | VDD_AL_1 | VDD_AL_2 | VDD_AH_10 | VDD_AH_11 | Ref_rext_1 |
| F | Reserved_65 | Reserved_64 | Reserved_218 | VDD_AH_17 | VDD_AH_18 | VDD_IO_5 | VDD_AH_3 | VDD_AH_19 | VDD_AL_5 | VDD_AL_6 | VDD_AH_20 | VDD_AH_21 | Reserved_219 |
| G | Reserved_67 | Reserved_66 | VSS_3 | Reserved_15 | VSS_4 | VDD_1 | VDD_2 | VDD_3 | VDD_AL_9 | VDD_AL_10 | VDD_4 | VDD_5 | Reserved_247 |
| H | Reserved_69 | Reserved_68 | VSS_7 | Reserved_14 | VSS_8 | VDD_11 | VDD_12 | VDD_13 | VDD_14 | VDD_15 | VDD_16 | VDD_17 | Reserved_246 |
| J | Reserved_71 | Reserved_70 | VDD_AH_27 | VDD_AH_28 | VDD_AL_13 | VDD_AL_14 | VDD_AL_15 | Reserved_240 | Reserved_241 | Reserved_242 | Reserved_243 | Reserved_244 | Reserved_245 |
| K | Reserved_73 | Reserved_72 | VSS_11 | Ref_rext_2 | VDD_AL_19 | VDD_AL_20 | VDD_AL_21 | VSS_12 | VSS_13 | VSS_14 | VSS_15 | VSS_16 | VSS_17 |
| L | Reserved_75 | Reserved_74 | VSS_25 | Ref_filt_2 | VSS_26 | VDD_25 | VDD_26 | VSS_27 | VSS_28 | VSS_29 | VSS_30 | VSS_31 | VSS_32 |
| M | Reserved_77 | Reserved_76 | VDD_AH_31 | VDD_AH_32 | VDD_AH_33 | VDD_29 | VDD_30 | VSS_41 | VSS_42 | VSS_43 | VSS_44 | VSS_45 | VSS_46 |
| N | Reserved_79 | Reserved_78 | VSS_53 | VSS_54 | VSS_55 | VDD_33 | VDD_34 | VSS_56 | VSS_57 | VSS_58 | VSS_59 | VSS_60 | VSS_61 |
| P | Reserved_81 | Reserved_80 | VSS_71 | Reserved_24 | VDD_IO_7 | VDD_37 | VDD_38 | VSS_72 | VSS_73 | VSS_74 | VSS_75 | VSS_76 | VSS_77 |
| R | GPIO_31 | GPIO_30 | GPIO_29 | GPIO_28 | VDD_IO_8 | VDD_41 | VDD_42 | VSS_86 | VSS_87 | VSS_88 | VSS_89 | VSS_90 | VSS_91 |
| T | GPIO_27 | GPIO_26 | GPIO_25 | GPIO_24 | VDD_IO_9 | VDD_45 | VDD_46 | VSS_98 | VSS_99 | VSS_100 | VSS_101 | VSS_102 | VSS_103 |
| U | GPIO_23 | GPIO_22 | GPIO_21 | GPIO_20 | VDD_IO_10 | VSS_110 | VSS_111 | VSS_112 | VSS_113 | VSS_114 | VSS_115 | VSS_116 | VSS_117 |
| V | GPIO_19 | GPIO_18 | GPIO_17 | GPIO_16 | VDD_IO_11 | VDD_49 | VDD_50 | VDD_51 | VDD_52 | VDD_53 | VDD_54 | VDD_55 | VDD_56 |
| W | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | VDD_IO_12 | VDD_65 | VDD_66 | VDD_67 | VDD_68 | VDD_69 | VDD_70 | VDD_71 | VDD_72 |
| Y | GPIO_11 | GPIO_10 | GPIO_9 | GPIO_8 | VDD_IO_13 | Reserved_146 | Reserved_141 | RefClk_P | Reserved_137 | Reserved_134 | Reserved_129 | VSS_126 | Reserved_126 |
| AA | GPIO_7 | GPIO_6 | GPIO_5 | GPIO_4 | VDD_IO_14 | Reserved_147 | Reserved_140 | RefClk_N | Reserved_136 | Reserved_135 | Reserved_128 | VSS_145 | Reserved_127 |
| AB | GPIO_3 | GPIO_2 | GPIO_1 | GPIO_0 | VDD_IO_15 | VSS_129 | VSS_130 | VSS_131 | VSS_132 | VSS_133 | VSS_134 | VSS_135 | VSS_136 |
| AC | SI_DO | SI_nEn | VSS_148 | VDD_IO_16 | VDD_IO_17 | VDD_A_1 | VDD_A_2 | VDD_A_3 | VDD_A_4 | VDD_A_5 | VDD_A_6 | VDD_A_7 | VDD_A_8 |
| AD | SI_Clk | SI_DI | Reserved_18 | VDD_IO_18 | VSS_149 | VDD_VS_1 | VDD_VS_2 | VDD_VS_3 | VDD_VS_4 | VDD_VS_5 | VDD_VS_6 | VDD_VS_7 | VDD_VS_8 |
| AE | VSS_151 | Reserved_17 | VDD_IO_19 | VSS_163 | VSS_152 | Reserved_144 | Reserved_143 | Reserved_22 | Reserved_139 | Reserved_132 | Reserved_131 | VSS_153 | Reserved_124 |
| AF | | VDD_IO_20 | MDIO | MDC | VSS_158 | Reserved_145 | Reserved_142 | Reserved_23 | Reserved_138 | Reserved_133 | Reserved_130 | VSS_159 | Reserved_125 |

***Figure 105*** • **Pin Diagram for VSC7424-02, Top Right**

| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P5_D0P | P5_D1P | P5_D2P | P5_D3P | P4_D0P | P4_D1P | P4_D2P | P4_D3P | P3_D0P | P3_D1P | P3_D2P | P3_D3P | | A |
| P5_D0N | P5_D1N | P5_D2N | P5_D3N | P4_D0N | P4_D1N | P4_D2N | P4_D3N | P3_D0N | P3_D1N | P3_D2N | P3_D3N | VSS_2 | B |
| RefClk_Sel2 | Reserved_8 | Reserved_7 | Reserved_6 | Reserved_5 | Reserved_201 | Reserved_202 | Reserved_203 | THERMDC_VSS | THERMDA | Reserved_204 | P2_D0N | P2_D0P | C |
| Reserved_12 | Reserved_212 | VDD_AH_5 | JTAG_CLK | JTAG_DI | JTAG_DO | JTAG_TMS | JTAG_TRST | Reserved_213 | Reserved_214 | Reserved_215 | P2_D1N | P2_D1P | D |
| Ref_filt_1 | VDD_AH_12 | VDD_AH_13 | VDD_AL_3 | VDD_AL_4 | VDD_AH_14 | VDD_IO_3 | VDD_IO_4 | VDD_AH_15 | VDD_AH_16 | Reserved_217 | P2_D2N | P2_D2P | E |
| Reserved_220 | VDD_AH_22 | VDD_AH_23 | VDD_AL_7 | VDD_AL_8 | VDD_AH_24 | VDD_AH_6 | VDD_IO_6 | VDD_AH_25 | VDD_AH_26 | Reserved_221 | P2_D3N | P2_D3P | F |
| Reserved_223 | VDD_6 | VDD_7 | VDD_AL_11 | VDD_AL_12 | VDD_8 | VDD_9 | VDD_10 | VSS_5 | Reserved_10 | VSS_6 | P1_D0N | P1_D0P | G |
| Reserved_225 | VDD_18 | VDD_19 | VDD_20 | VDD_21 | VDD_22 | VDD_23 | VDD_24 | VSS_9 | Reserved_11 | VSS_10 | P1_D1N | P1_D1P | H |
| Reserved_232 | Reserved_233 | Reserved_234 | Reserved_235 | Reserved_236 | Reserved_237 | VDD_AL_16 | VDD_AL_17 | VDD_AL_18 | VDD_AH_29 | VDD_AH_30 | P1_D2N | P1_D2P | J |
| VSS_18 | VSS_19 | VSS_20 | VSS_21 | VSS_22 | VSS_23 | VDD_AL_22 | VDD_AL_23 | VDD_AL_24 | Ref_rext_0 | VSS_24 | P1_D3N | P1_D3P | K |
| VSS_33 | VSS_34 | VSS_35 | VSS_36 | VSS_37 | VSS_38 | VDD_27 | VDD_28 | VSS_39 | Ref_filt_0 | VSS_40 | P0_D0N | P0_D0P | L |
| VSS_47 | VSS_48 | VSS_49 | VSS_50 | VSS_51 | VSS_52 | VDD_31 | VDD_32 | VDD_AH_34 | VDD_AH_35 | VDD_AH_36 | P0_D1N | P0_D1P | M |
| VSS_62 | VSS_63 | VSS_64 | VSS_65 | VSS_66 | VSS_67 | VDD_35 | VDD_36 | VSS_68 | VSS_69 | VSS_70 | P0_D2N | P0_D2P | N |
| VSS_78 | VSS_79 | VSS_80 | VSS_81 | VSS_82 | VSS_83 | VDD_39 | VDD_40 | VDD_IODDR_1 | VSS_84 | VSS_85 | P0_D3N | P0_D3P | P |
| VSS_92 | VSS_93 | VSS_94 | VSS_95 | VSS_96 | VSS_97 | VDD_43 | VDD_44 | VDD_IODDR_2 | Reserved_20 | Reserved_19 | DDR_Rext | DDR_Vref | R |
| VSS_104 | VSS_105 | VSS_106 | VSS_107 | VSS_108 | VSS_109 | VDD_47 | VDD_48 | VDD_IODDR_3 | Reserved_21 | DDR_A13 | DDR_A12 | DDR_A11 | T |
| VSS_118 | VSS_119 | VSS_120 | VSS_121 | VSS_122 | VSS_123 | VSS_124 | VSS_125 | VDD_IODDR_4 | DDR_A7 | DDR_A9 | DDR_A6 | DDR_A8 | U |
| VDD_57 | VDD_58 | VDD_59 | VDD_60 | VDD_61 | VDD_62 | VDD_63 | VDD_64 | VDD_IODDR_5 | DDR_A3 | DDR_A5 | DDR_A2 | DDR_A4 | V |
| VDD_73 | VDD_74 | VDD_75 | VDD_76 | VDD_77 | VDD_78 | VDD_79 | VDD_80 | VDD_IODDR_6 | DDR_A10 | DDR_A1 | DDR_nCAS | DDR_A0 | W |
| Reserved_121 | Reserved_118 | VSS_127 | Reserved_113 | SerDes1_TxP | SerDes0_TxP | VSS_128 | Reserved_102 | VDD_IODDR_7 | DDR_BA0 | DDR_BA1 | DDR_ODT | DDR_nRAS | Y |
| Reserved_120 | Reserved_119 | VSS_146 | Reserved_112 | SerDes1_TxN | SerDes0_TxN | VSS_147 | Reserved_103 | VDD_IODDR_8 | DDR_nWE | DDR_BA2 | DDR_CK | DDR_CKn | AA |
| VSS_137 | VSS_138 | VSS_139 | VSS_140 | VSS_141 | VSS_142 | VSS_143 | VSS_144 | VDD_IODDR_9 | DDR_DQ3 | DDR_CKE | DDR_DQ2 | DDR_DQ5 | AB |
| VDD_A_9 | VDD_A_10 | VDD_A_11 | VDD_A_12 | VDD_A_13 | VDD_A_14 | VDD_A_15 | VDD_A_16 | VDD_IODDR_10 | DDR_DQ1 | DDR_DQ4 | DDR_DQ7 | DDR_DQ0 | AC |
| VDD_VS_9 | VDD_VS_10 | VDD_VS_11 | VDD_VS_12 | VDD_VS_13 | VDD_VS_14 | VDD_VS_15 | VDD_VS_16 | VSS_150 | VDD_IODDR_11 | DDR_DQ6 | DDR_DQS | DDR_DQSn | AD |
| Reserved_123 | Reserved_116 | VSS_154 | Reserved_115 | SerDes1_RxP | SerDes0_RxP | VSS_155 | Reserved_100 | SerDes_Rext_0 | VSS_156 | VDD_IODDR_12 | DDR_DM | VSS_157 | AE |
| Reserved_122 | Reserved_117 | VSS_160 | Reserved_114 | SerDes1_RxN | SerDes0_RxN | VSS_161 | Reserved_101 | SerDes_Rext_1 | VSS_162 | VDD_IODDR_14 | VDD_IODDR_13 | | AF |

# 9.2 Pins by Function for VSC7424-02

This section contains the functional pin descriptions for the VSC7424-02 device. The following table lists the definitions for the pin type symbols.

***Table 854*** • **Pin Type Symbol Definitions**

| Symbol | Pin Type | Description |
|---|---|---|
| ABIAS | Analog bias | Analog bias pin. |
| DIFF | Differential | Differential signal pair. |
| I | Input | Input signal. |
| O | Output | Output signal. |
| I/O | Bidirectional | Bidirectional input or output signal. |
| A | Analog input | Analog input for sensing variable voltage levels. |
| PD | Pull-down | On-chip pull-down resistor to VSS. |
| PU | Pull-up | On-chip pull-up resistor to VDD_IO. |
| 3V | | 3.3 V-tolerant. |
| O | Output | Output signal. |

*Table 854 •* **Pin Type Symbol Definitions *(continued)***

| Symbol | Pin Type | Description |
|---|---|---|
| OZ | 3-state output | Output. |
| ST | Schmitt-trigger | Input has Schmitt-trigger circuitry. |
| TD | Termination differential | Internal differential termination. |

## 9.2.1 Analog Bias Signals

The following table lists the pins associated with the analog bias signals.

*Table 855 •* **Analog Bias Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_Rext_[1:0] | A | Analog bias calibration.<br>Connect an external 620 Ω ±1% resistor between SerDes_Rext_1 and SerDes_Rext_0. |
| Ref_filt_[2:0] | A | Reference filter. Connect a 1.0 µF external capacitor between each pin and ground. |
| Ref_rext_[2:0] | A | Reference external resistor. Connect a 2.0 kΩ (1%) resistor between each pin and ground. |

## 9.2.2 Clock Circuits

The following table lists the pins associated with the system clock interface.

*Table 856 •* **System Clock Interface Pins**

| Name | Type | Description |
|---|---|---|
| RefClk_Sel[2:0] | I, PD | Reference clock frequency selection.<br>0: Connect to pull-down or leave floating.<br>1: Connect to pull-up to $V_{DD_IO}$.<br>Coding:<br>000: 125 MHz (default).<br>001: 156.25 MHz.<br>010: Reserved.<br>011: Reserved.<br>100: 25 MHz.<br>101: Reserved.<br>110: Reserved.<br>111: Reserved. |
| RefClk_P<br>RefClk_N | I, Diff | Reference clock input.<br>The input can be either differential or single-ended.<br>In differential mode, REFCLK_P is the true part of the differential signal, and REFCLK_N is the complement part of the differential signal.<br>In single-ended mode, REFCLK_P is used as single-ended LVTTL input, and the REFCLK_N should be pulled to $V_{DD_A}$.<br>Required applied frequency depends on RefClk_Sel[2:0] input state. See description for RefClk_Sel[2:0] pins. |

### 9.2.3 DDR2 SDRAM Interface

The following table lists the pins associated with the DDR2 SDRAM interface.

***Table 857 •*** **DDR2 SDRAM Pins**

| Name | Type | Description |
|---|---|---|
| DDR_CK<br>DDR_CKn | 0, Diff | SDRAM differential clock. Differential clock to external SDRAM.<br>DDR_CK is the true part of the differential signal.<br>DDR_nCk is the complement part. |
| DDR_CKE | O | SDRAM clock enable.<br>0: Disables clock in external SDRAM.<br>1: Enables clock in external SDRAM. |
| DDR_nRAS<br>DDR_nCAS<br>DDR_nWE | O | SDRAM command outputs.<br>DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered. |
| DDR_DM | O | SDRAM data mask outputs.<br>DDR_DM and DDR_UDM are mask signals for data written to the SDRAM. DDR_LDM corresponds to data on DDR_DQ[7:0], and DDR_UDM corresponds to data on DDR_DQ[15:8]. |
| DDR_BA[2:0] | O | SDRAM bank address outputs.<br>DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied. |
| DDR_A[13:0] | O | SDRAM address outputs.<br>Provide row and column addresses to the SDRAM. |
| DDR_DQ[7:0] | I/O | SDRAM data bus. |
| DDR_DQS<br>DDR_DQSn | I/O, Diff | SDRAM differential data strobes.<br>Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. |
| DDR_ODT | O | Control signals for the attached DDR2 SDRAM devices on-die termination. |
| DDR_Rext | ABIAS | External DDR impedance calibration.<br>Connect the pin through an external 240 Ω ±1% resistor to ground. |
| DDR_Vref | ABIAS | Input reference voltage.<br>Provides the input switching reference voltage to the SSTL DDR signals. |

### 9.2.4 General-Purpose Inputs and Outputs

The following table lists the pins associated with general-purpose inputs and outputs. The GPIO pins have an alternate function signal, which is mapped out in the following table. The overlaid functions are

selected by software on a pin-by-pin basis. The parallel interface and MIIM slave interface are enabled depending on the VCORE_CFG settings and override the normal GPIO and alternate functions.

***Table 858 •*** **GPIO Pin Mapping**

| Name | Overlaid Function 1 | Overlaid Function 2 | Parallel Interface | MIIM Slave Interface | Type |
|---|---|---|---|---|---|
| GPIO_0 | SIO_CLK | | | | I/O, PU,ST, 3V |
| GPIO_1 | SIO_LD | | | | I/O, PU,ST, 3V |
| GPIO_2 | SIO_DO | | | | I/O, PU,ST, 3V |
| GPIO_3 | SIO_DI | | | | I/O, PU,ST, 3V |
| GPIO_4 | TACHO | | | | I/O, PU,ST, 3V |
| GPIO_5 | TWI_SCL | PHY0_LED1 | | | I/O, PU,ST, 3V |
| GPIO_6 | TWI_SDA | PHY1_LED1 | | | I/O, PU,ST, 3V |
| GPIO_7 | | PHY2_LED1 | | | I/O, PU,ST, 3V |
| GPIO_8 | EXT_IRQ0 | PHY3_LED1 | | | I/O, PU,ST, 3V |
| GPIO_9 | EXT_IRQ1 | PHY4_LED1 | | | I/O, PU,ST, 3V |
| GPIO_10 | SFP14_SD | PHY5_LED1 | | | I/O, PU,ST, 3V |
| GPIO_11 | SFP15_SD | PHY6_LED1 | | | I/O, PU,ST, 3V |
| GPIO_12 | SFP17_SD | PHY7_LED1 | | | I/O, PU,ST, 3V |
| GPIO_13 | SFP18_SD | PHY8_LED1 | PI_nCS | | I/O, PU,ST, 3V |
| GPIO_14 | SI_nEN1 | PHY9_LED1 | PI_nWR | SLV_ADDR | I/O, PU,ST, 3V |
| GPIO_15 | SI_nEn2 | PHY10_LED1 | PI_nOE | SLV_MDC | I/O, PU,ST, 3V |
| GPIO_16 | SI_nEn3 | PHY11_LED1 | PI_nDone | SLV_MDIO | I/O, PU,ST, 3V |
| GPIO_17 | SFP10_SD | PHY0_LED0 | PI_A0 | | I/O, PU,ST, 3V |
| GPIO_18 | SFP11_SD | PHY2_LED0 | PI_A1 | | I/O, PU,ST, 3V |
| GPIO_19 | SFP12_SD | PHY2_LED0 | PI_A2 | | I/O, PU,ST, 3V |
| GPIO_20 | SFP13_SD | PHY3_LED0 | PI_A3 | | I/O, PU,ST, 3V |
| GPIO_21 | SFP16_SD | PHY4_LED0 | PI_D0 | | I/O, PU,ST, 3V |
| GPIO_22 | SFP19_SD | PHY5_LED0 | PI_D1 | | I/O, PU,ST, 3V |
| GPIO_23 | SFP24_SD | PHY6_LED0 | PI_D2 | | I/O, PU,ST, 3V |
| GPIO_24 | SFP25_SD | PHY7_LED0 | PI_D3 | | I/O, PU,ST, 3V |
| GPIO_25 | SFP20_SD | PHY8_LED0 | PI_D4 | | I/O, PU,ST, 3V |
| GPIO_26 | SFP21_SD | PHY9_LED0 | PI_D5 | | I/O, PU,ST, 3V |
| GPIO_27 | SFP22_SD | PHY10_LED0 | PI_D6 | | I/O, PU,ST, 3V |
| GPIO_28 | SFP23_SD | PHY11_LED0 | PI_D7 | | I/O, PU,ST, 3V |
| GPIO_29 | PWM | | | | I/O, PU,ST, 3V |
| GPIO_30 | UART_TX | | | | I/O, PU,ST, 3V |
| GPIO_31 | UART_RX | | | | I/O, PU,ST, 3V |

## 9.2.5 JTAG Interface

The following table lists the pins associated with the JTAG interface. The JTAG interface can be connected to the boundary scan TAP controller or the internal VCore-III TAP controller for software debug as described for the VCore_ICE_nEn signal.

The JTAG signals are not 5 V tolerant.

***Table 859 •*** **JTAG Interface Pins**

| Name | Type | Description |
|---|---|---|
| JTAG_nTRST | I, PU, ST, 3V | JTAG test reset, active low. For normal device operation, JTAG_nTRST should be pulled low. |
| JTAG_TCK | I, PU, ST, 3V | JTAG clock. |
| JTAG_TDI | I, PU, ST, 3V | JTAG test data in. |
| JTAG_TDO | OZ, 3V | JTAG test data out. |
| JTAG_TMS | I, PU, ST, 3V | JTAG test mode select. |

## 9.2.6 MII Management Interface

The following table lists the pins associated with the MII Management interface.

***Table 860 •*** **MII Management Interface Pins**

| Name | Type | Description |
|---|---|---|
| MDIO | I/O, 3V | Management data input/output.<br>MDIO is a bidirectional signal between a PHY and the device, used to transfer control and status information. Control information is driven by the device synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by the device. |
| MDC | O, 3V | Management data clock.<br>MDC is sourced by the station management entity (the device) to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal. |

## 9.2.7 Miscellaneous Signals

The following table lists the pins associated with a particular interface or facility on the device.

***Table 861 •*** **Miscellaneous Pins**

| Name | Type | Description |
|---|---|---|
| nReset | I, PD, ST, 3V | Global device reset, active low. |
| COMA_MODE | I/O, PU, ST, 3V | When this pin is asserted high, all PHYs are held in a powered-down state.<br>When this pin is deasserted low, all PHYs are powered up and resume normal operation. Additionally, this signal is used to synchronize the operation of multiple devices on the same printed circuit board to provide visual synchronization for LEDs driven from the separate devices. |

*Table 861 •* **Miscellaneous Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| VCORE_CFG[2:0] | I, PD, ST, 3V | Configuration signals for controlling the VCore-III CPU functions. |
| VCore_ICE_nEn | I, PU, 3V | VCore ICE nEn.<br>0: Enables the VCore-III JTAG debug interface over the JTAG interface pins.<br>1: Enables normal IO-JTAG over the JTAG interface. |
| THERMDA | A | Thermal diode anode (p-junction). |
| THERMDC_VSS | A | Thermal diode cathode (n-junction). Connected on-die to $V_{SS}$. |
| EXT_IRQ[1:0](1) | I/O PD, 3V | This pin interrupts inputs or outputs to the internal VCore-III CPU system or to an external processor. Signal polarity is programmable. |
| Reserved_1<br>Reserved_[5:7] | I, PD, ST, 3V | Tie to $V_{DD_IO}$. |
| Reserved_4<br>Reserved_8 | I, PD, ST, 3V | Tie to $V_{SS}$. |
| Reserved_[10:15]<br>Reserved_[17:24]<br>Reserved_[50:81]<br>Reserved_[100:103]<br>Reserved_[112:147]<br>Reserved_[201:209]<br>Reserved_[211:221]<br>Reserved_[223]<br>Reserved_[225]<br>Reserved_[232:237]<br>Reserved_[240:248] | I, PD, ST, 3V | Leave floating. |

1. Available as an alternate function on the GPIO_8 and GPIO_9 pins.

## 9.2.8 Parallel Interface

The parallel interface (PI) can operate in a Master mode or a Slave mode according to the VCORE_CFG[1:0] signal settings. In Master mode, the internal VCore-III CPU system controls the PI and can access external peripherals over it. In Slave mode, the PI can be used by an external CPU to access internal device resources.

The PI master and slave mode signals are alternate function signals on GPIO pins. For more information about the GPIO mapping, see Table 858, page 641.

*Table 862 •* **Parallel Interface VCore-III Master Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | OZ, 3V | External address bus.<br>Used for addressing external memory space. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | External data bus. PI_Data0 is LSB. |
| PI_nCS | OZ, 3V | Programmable active low chip selects.<br>PI_nCS is used as default for booting from external memory (typically Flash). |

*Table 862 •* **Parallel Interface VCore-III Master Mode Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| PI_nDone | I, 3V | Acknowledges an operation. Used for external device-paced access operation.<br>Signal polarity is programmable. |
| PI_nOE | OZ, 3V | Active low signal that signals external device to drive data bus during read access. |
| PI_nWR | OZ, 3V | Active low signal that signals external access direction. Read (1) or write (0). |

The following pins are associated with the parallel CPU interface slave mode.

*Table 863 •* **Parallel CPU Interface Slave Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | I, 3V | Internal device register address bus. Controlled by external CPU. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | Data bus.<br>PI_Data[0] is LSB. |
| PI_nCS | I, 3V | Device chip select. |
| PI_nDone | O, 3V | Acknowledges an operation. Signal polarity is programmable. |
| PI_nOE | I, 3V | Signals device to drive data bus during read operations. |
| PI_nWR | I, 3V | Signals access direction. Read (1) or write (0). |

## 9.2.9 Power Supplies and Ground

The following table lists the power supply and ground pins.

*Table 864 •* **Power Supply and Ground Pins**

| Name | Type | Description |
|---|---|---|
| VDD | Power | 1.0 V power supply voltage for core |
| VDD_A | Power | 1.0 V power supply voltage for analog circuits |
| VDD_AL | Power | 1.0 V power supply voltage for analog circuits for twisted pair interface |
| VDD_AH | Power | 2.5 V power supply voltage for analog driver in twisted pair interface |
| VDD_IO | Power | 2.5 V power supply for MII Management interface, parallel CPU interface, and miscellaneous I/Os |
| VDD_IODDR | Power | 1.8 V power supply for DDR interface |
| VDD_VS | Power | 1.0 V or 1.2 V power supply for SerDes and Enhanced SerDes interfaces |
| VSS | Ground | Ground reference |

## 9.2.10 Serial CPU Interface

The serial CPU interface (SI) can be used as a serial slave, master, or boot interface.

As a slave interface, it allows external CPUs to access internal registers. As a master interface, it allows the device to access external devices using a programmable protocol. The serial CPU interface also allows the internal VCore-III CPU system to boot from an attached serial memory device when the VCORE_CFG signals are set appropriately.

The following table lists the pins associated with the serial CPU interface (SI).

***Table 865 •*** **Serial CPU Interface Pins**

| Name | Type | Description |
|---|---|---|
| SI_Clk | I/O, 3V | Slave mode: Input receiving serial interface clock from external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven with clock to external serial memory device. |
| SI_DI | I, 3V | Slave mode: Input receiving serial interface data from external master.<br>Master mode: Input directly read by software from register bit.<br>Boot mode: Input boot data from external serial memory device. |
| SI_DO | OZ, 3V | Slave mode: Output transmitting serial interface data to external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: No function. |
| SI_nEn<br>SI_nEn[3:1](1) | I/O, 3V | Slave mode: Input used to enable SI slave interface.<br>0: Enabled<br>1: Disabled<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven while booting from EEPROM or serial flash to internal VCore-III CPU system. Released when booting is completed. |

1. Available as an alternate function on the GPIO_16, GPIO_15, and GPIO_14 pins. For more information about GPIO pin mapping, see Table 858, page 641.

## 9.2.11 SerDes Interface

The following pins are associated with the SerDes (SGMII) interface.

***Table 866 •*** **SerDes Interface Pins**

| Name | Type | Description |
|---|---|---|
| SerDes[1:0]_RxP, N | I, Diff, TD | Differential SerDes data inputs. |
| SerDes[1:0]_TxP, N | O, Diff | Differential SerDes data outputs. |

## 9.2.12 Twisted Pair Interface

The following pins are associated with the twisted pair interface. The PHYn_LED[1:0] LED control signals associated with the twisted pair interfaces are alternate functions on the GPIOs. For more information about the GPIO pin mapping, see Table 858, page 641.

***Table 867 •*** **Twisted Pair Interface Pins**

| Name | Type | Description |
|---|---|---|
| P0_D0P<br>P1_D0P<br>P2_D0P<br>P3_D0P<br>P4_D0P<br>P5_D0P<br>P6_D0P<br>P7_D0P | $A_{DIFF}$ | Tx/Rx channel A positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1. |
| P0_D0N<br>P1_D0N<br>P2_D0N<br>P3_D0N<br>P4_D0N<br>P5_D0N<br>P6_D0N<br>P7_D0N | $A_{DIFF}$ | Tx/Rx channel A negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2. |
| P0_D1P<br>P1_D1P<br>P2_D1P<br>P3_D1P<br>P4_D1P<br>P5_D1P<br>P6_D1P<br>P7_D1P | $A_{DIFF}$ | Tx/Rx channel B positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3. |
| P0_D1N<br>P1_D1N<br>P2_D1N<br>P3_D1N<br>P4_D1N<br>P5_D1N<br>P6_D1N<br>P7_D1N | $A_{DIFF}$ | Tx/Rx channel B negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6. |
| P0_D2P<br>P1_D2P<br>P2_D2P<br>P3_D2P<br>P4_D2P<br>P5_D2P<br>P6_D2P<br>P7_D2P | $A_{DIFF}$ | Tx/Rx channel C positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes). |

*Table 867 •* **Twisted Pair Interface Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| P0_D2N<br>P1_D2N<br>P2_D2N<br>P3_D2N<br>P4_D2N<br>P5_D2N<br>P6_D2N<br>P7_D2N | $A_{DIFF}$ | Tx/Rx channel C negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes). |
| P0_D3P<br>P1_D3P<br>P2_D3P<br>P3_D3P<br>P4_D3P<br>P5_D3P<br>P6_D3P<br>P7_D3P | $A_{DIFF}$ | Tx/Rx channel D positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes). |
| P0_D3N<br>P1_D3N<br>P2_D3N<br>P3_D3N<br>P4_D3N<br>P5_D3N<br>P6_D3N<br>P7_D3N | $A_{DIFF}$ | Tx/Rx channel D negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes). |

## 9.3 Pins by Number for VSC7424-02

This section provides a numeric list of the VSC7424-02 pins.

| | |
|---|---|
| A2 | Reserved_57 |
| A3 | Reserved_55 |
| A4 | Reserved_53 |
| A5 | Reserved_51 |
| A6 | P7_D0P |
| A7 | P7_D1P |
| A8 | P7_D2P |
| A9 | P7_D3P |
| A10 | P6_D0P |
| A11 | P6_D1P |
| A12 | P6_D2P |
| A13 | P6_D3P |
| A14 | P5_D0P |
| A15 | P5_D1P |
| A16 | P5_D2P |
| A17 | P5_D3P |
| A18 | P4_D0P |
| A19 | P4_D1P |
| A20 | P4_D2P |
| A21 | P4_D3P |
| A22 | P3_D0P |
| A23 | P3_D1P |
| A24 | P3_D2P |
| A25 | P3_D3P |
| AA1 | GPIO_7 |
| AA2 | GPIO_6 |
| AA3 | GPIO_5 |
| AA4 | GPIO_4 |
| AA5 | VDD_IO_14 |
| AA6 | Reserved_147 |
| AA7 | Reserved_140 |
| AA8 | RefClk_N |
| AA9 | Reserved_136 |
| AA10 | Reserved_135 |
| AA11 | Reserved_128 |
| AA12 | VSS_145 |
| AA13 | Reserved_127 |
| AA14 | Reserved_120 |
| AA15 | Reserved_119 |
| AA16 | VSS_146 |
| AA17 | Reserved_112 |
| AA18 | SerDes1_TxN |
| AA19 | SerDes0_TxN |
| AA20 | VSS_147 |
| AA21 | Reserved_103 |
| AA22 | VDD_IODDR_8 |
| AA23 | DDR_nWE |
| AA24 | DDR_BA2 |
| AA25 | DDR_CK |
| AA26 | DDR_CKn |
| AB1 | GPIO_3 |
| AB2 | GPIO_2 |
| AB3 | GPIO_1 |
| AB4 | GPIO_0 |
| AB5 | VDD_IO_15 |
| AB6 | VSS_129 |
| AB7 | VSS_130 |
| AB8 | VSS_131 |
| AB9 | VSS_132 |
| AB10 | VSS_133 |
| AB11 | VSS_134 |
| AB12 | VSS_135 |
| AB13 | VSS_136 |
| AB14 | VSS_137 |
| AB15 | VSS_138 |
| AB16 | VSS_139 |
| AB17 | VSS_140 |
| AB18 | VSS_141 |
| AB19 | VSS_142 |
| AB20 | VSS_143 |
| AB21 | VSS_144 |
| AB22 | VDD_IODDR_9 |
| AB23 | DDR_DQ3 |
| AB24 | DDR_CKE |
| AB25 | DDR_DQ2 |
| AB26 | DDR_DQ5 |
| AC1 | SI_DO |
| AC2 | SI_nEn |
| AC3 | VSS_148 |
| AC4 | VDD_IO_16 |
| AC5 | VDD_IO_17 |
| AC6 | VDD_A_1 |
| AC7 | VDD_A_2 |
| AC8 | VDD_A_3 |
| AC9 | VDD_A_4 |
| AC10 | VDD_A_5 |
| AC11 | VDD_A_6 |
| AC12 | VDD_A_7 |
| AC13 | VDD_A_8 |
| AC14 | VDD_A_9 |
| AC15 | VDD_A_10 |
| AC16 | VDD_A_11 |
| AC17 | VDD_A_12 |
| AC18 | VDD_A_13 |
| AC19 | VDD_A_14 |
| AC20 | VDD_A_15 |
| AC21 | VDD_A_16 |
| AC22 | VDD_IODDR_10 |
| AC23 | DDR_DQ1 |
| AC24 | DDR_DQ4 |
| AC25 | DDR_DQ7 |
| AC26 | DDR_DQ0 |
| AD1 | SI_Clk |
| AD2 | SI_DI |
| AD3 | Reserved_18 |
| AD4 | VDD_IO_18 |
| AD5 | VSS_149 |
| AD6 | VDD_VS_1 |
| AD7 | VDD_VS_2 |
| AD8 | VDD_VS_3 |

Pins by number *(continued)*

| | |
|---|---|
| AD9 | VDD_VS_4 |
| AD10 | VDD_VS_5 |
| AD11 | VDD_VS_6 |
| AD12 | VDD_VS_7 |
| AD13 | VDD_VS_8 |
| AD14 | VDD_VS_9 |
| AD15 | VDD_VS_10 |
| AD16 | VDD_VS_11 |
| AD17 | VDD_VS_12 |
| AD18 | VDD_VS_13 |
| AD19 | VDD_VS_14 |
| AD20 | VDD_VS_15 |
| AD21 | VDD_VS_16 |
| AD22 | VSS_150 |
| AD23 | VDD_IODDR_11 |
| AD24 | DDR_DQ6 |
| AD25 | DDR_DQS |
| AD26 | DDR_DQSn |
| AE1 | VSS_151 |
| AE2 | Reserved_17 |
| AE3 | VDD_IO_19 |
| AE4 | VSS_163 |
| AE5 | VSS_152 |
| AE6 | Reserved_144 |
| AE7 | Reserved_143 |
| AE8 | Reserved_22 |
| AE9 | Reserved_139 |
| AE10 | Reserved_132 |
| AE11 | Reserved_131 |
| AE12 | VSS_153 |
| AE13 | Reserved_124 |
| AE14 | Reserved_123 |
| AE15 | Reserved_116 |
| AE16 | VSS_154 |
| AE17 | Reserved_115 |
| AE18 | SerDes1_RxP |
| AE19 | SerDes0_RxP |
| AE20 | VSS_155 |
| AE21 | Reserved_100 |
| AE22 | SerDes_Rext_0 |
| AE23 | VSS_156 |
| AE24 | VDD_IODDR_12 |
| AE25 | DDR_DM |
| AE26 | VSS_157 |
| AF2 | VDD_IO_20 |
| AF3 | MDIO |
| AF4 | MDC |
| AF5 | VSS_158 |
| AF6 | Reserved_145 |
| AF7 | Reserved_142 |
| AF8 | Reserved_23 |
| AF9 | Reserved_138 |
| AF10 | Reserved_133 |
| AF11 | Reserved_130 |
| AF12 | VSS_159 |
| AF13 | Reserved_125 |
| AF14 | Reserved_122 |
| AF15 | Reserved_117 |
| AF16 | VSS_160 |
| AF17 | Reserved_114 |
| AF18 | SerDes1_RxN |
| AF19 | SerDes0_RxN |
| AF20 | VSS_161 |
| AF21 | Reserved_101 |
| AF22 | SerDes_Rext_1 |
| AF23 | VSS_162 |
| AF24 | VDD_IODDR_14 |
| AF25 | VDD_IODDR_13 |
| B1 | VSS_1 |
| B2 | Reserved_56 |
| B3 | Reserved_54 |
| B4 | Reserved_52 |
| B5 | Reserved_50 |
| B6 | P7_D0N |
| B7 | P7_D1N |
| B8 | P7_D2N |
| B9 | P7_D3N |
| B10 | P6_D0N |
| B11 | P6_D1N |
| B12 | P6_D2N |
| B13 | P6_D3N |
| B14 | P5_D0N |
| B15 | P5_D1N |
| B16 | P5_D2N |
| B17 | P5_D3N |
| B18 | P4_D0N |
| B19 | P4_D1N |
| B20 | P4_D2N |
| B21 | P4_D3N |
| B22 | P3_D0N |
| B23 | P3_D1N |
| B24 | P3_D2N |
| B25 | P3_D3N |
| B26 | VSS_2 |
| C1 | Reserved_59 |
| C2 | Reserved_58 |
| C3 | COMA_MODE |
| C4 | nRESET |
| C5 | VDD_IO_21 |
| C6 | Reserved_1 |
| C7 | VCORE_CFG0 |
| C8 | VCORE_CFG1 |
| C9 | VCORE_CFG2 |
| C10 | VCore_ICE_nEn |
| C11 | Reserved_4 |
| C12 | RefClk_Sel0 |
| C13 | RefClk_Sel1 |
| C14 | RefClk_Sel2 |
| C15 | Reserved_8 |
| C16 | Reserved_7 |
| C17 | Reserved_6 |
| C18 | Reserved_5 |
| C19 | Reserved_201 |
| C20 | Reserved_202 |
| C21 | Reserved_203 |
| C22 | THERMDC_VSS |
| C23 | THERMDA |

Pins by number *(continued)*

| | |
|---|---|
| C24 | Reserved_204 |
| C25 | P2_D0N |
| C26 | P2_D0P |
| D1 | Reserved_61 |
| D2 | Reserved_60 |
| D3 | Reserved_205 |
| D4 | VDD_AH_1 |
| D5 | VDD_AH_2 |
| D6 | Reserved_206 |
| D7 | Reserved_207 |
| D8 | Reserved_208 |
| D9 | Reserved_209 |
| D10 | Reserved_248 |
| D11 | VDD_AH_4 |
| D12 | Reserved_211 |
| D13 | Reserved_13 |
| D14 | Reserved_12 |
| D15 | Reserved_212 |
| D16 | VDD_AH_5 |
| D17 | JTAG_CLK |
| D18 | JTAG_DI |
| D19 | JTAG_DO |
| D20 | JTAG_TMS |
| D21 | JTAG_TRST |
| D22 | Reserved_213 |
| D23 | Reserved_214 |
| D24 | Reserved_215 |
| D25 | P2_D1N |
| D26 | P2_D1P |
| E1 | Reserved_63 |
| E2 | Reserved_62 |
| E3 | Reserved_216 |
| E4 | VDD_AH_7 |
| E5 | VDD_AH_8 |
| E6 | VDD_IO_1 |
| E7 | VDD_IO_2 |
| E8 | VDD_AH_9 |
| E9 | VDD_AL_1 |
| E10 | VDD_AL_2 |
| E11 | VDD_AH_10 |
| E12 | VDD_AH_11 |
| E13 | Ref_rext_1 |
| E14 | Ref_filt_1 |
| E15 | VDD_AH_12 |
| E16 | VDD_AH_13 |
| E17 | VDD_AL_3 |
| E18 | VDD_AL_4 |
| E19 | VDD_AH_14 |
| E20 | VDD_IO_3 |
| E21 | VDD_IO_4 |
| E22 | VDD_AH_15 |
| E23 | VDD_AH_16 |
| E24 | Reserved_217 |
| E25 | P2_D2N |
| E26 | P2_D2P |
| F1 | Reserved_65 |
| F2 | Reserved_64 |
| F3 | Reserved_218 |
| F4 | VDD_AH_17 |
| F5 | VDD_AH_18 |
| F6 | VDD_IO_5 |
| F7 | VDD_AH_3 |
| F8 | VDD_AH_19 |
| F9 | VDD_AL_5 |
| F10 | VDD_AL_6 |
| F11 | VDD_AH_20 |
| F12 | VDD_AH_21 |
| F13 | Reserved_219 |
| F14 | Reserved_220 |
| F15 | VDD_AH_22 |
| F16 | VDD_AH_23 |
| F17 | VDD_AL_7 |
| F18 | VDD_AL_8 |
| F19 | VDD_AH_24 |
| F20 | VDD_AH_6 |
| F21 | VDD_IO_6 |
| F22 | VDD_AH_25 |
| F23 | VDD_AH_26 |
| F24 | Reserved_221 |
| F25 | P2_D3N |
| F26 | P2_D3P |
| G1 | Reserved_67 |
| G2 | Reserved_66 |
| G3 | VSS_3 |
| G4 | Reserved_15 |
| G5 | VSS_4 |
| G6 | VDD_1 |
| G7 | VDD_2 |
| G8 | VDD_3 |
| G9 | VDD_AL_9 |
| G10 | VDD_AL_10 |
| G11 | VDD_4 |
| G12 | VDD_5 |
| G13 | Reserved_247 |
| G14 | Reserved_223 |
| G15 | VDD_6 |
| G16 | VDD_7 |
| G17 | VDD_AL_11 |
| G18 | VDD_AL_12 |
| G19 | VDD_8 |
| G20 | VDD_9 |
| G21 | VDD_10 |
| G22 | VSS_5 |
| G23 | Reserved_10 |
| G24 | VSS_6 |
| G25 | P1_D0N |
| G26 | P1_D0P |
| H1 | Reserved_69 |
| H2 | Reserved_68 |
| H3 | VSS_7 |
| H4 | Reserved_14 |
| H5 | VSS_8 |
| H6 | VDD_11 |
| H7 | VDD_12 |
| H8 | VDD_13 |
| H9 | VDD_14 |
| H10 | VDD_15 |

Pins by number *(continued)*

| | |
|---|---|
| H11 | VDD_16 |
| H12 | VDD_17 |
| H13 | Reserved_246 |
| H14 | Reserved_225 |
| H15 | VDD_18 |
| H16 | VDD_19 |
| H17 | VDD_20 |
| H18 | VDD_21 |
| H19 | VDD_22 |
| H20 | VDD_23 |
| H21 | VDD_24 |
| H22 | VSS_9 |
| H23 | Reserved_11 |
| H24 | VSS_10 |
| H25 | P1_D1N |
| H26 | P1_D1P |
| J1 | Reserved_71 |
| J2 | Reserved_70 |
| J3 | VDD_AH_27 |
| J4 | VDD_AH_28 |
| J5 | VDD_AL_13 |
| J6 | VDD_AL_14 |
| J7 | VDD_AL_15 |
| J8 | Reserved_240 |
| J9 | Reserved_241 |
| J10 | Reserved_242 |
| J11 | Reserved_243 |
| J12 | Reserved_244 |
| J13 | Reserved_245 |
| J14 | Reserved_232 |
| J15 | Reserved_233 |
| J16 | Reserved_234 |
| J17 | Reserved_235 |
| J18 | Reserved_236 |
| J19 | Reserved_237 |
| J20 | VDD_AL_16 |
| J21 | VDD_AL_17 |
| J22 | VDD_AL_18 |
| J23 | VDD_AH_29 |
| J24 | VDD_AH_30 |
| J25 | P1_D2N |
| J26 | P1_D2P |
| K1 | Reserved_73 |
| K2 | Reserved_72 |
| K3 | VSS_11 |
| K4 | Ref_rext_2 |
| K5 | VDD_AL_19 |
| K6 | VDD_AL_20 |
| K7 | VDD_AL_21 |
| K8 | VSS_12 |
| K9 | VSS_13 |
| K10 | VSS_14 |
| K11 | VSS_15 |
| K12 | VSS_16 |
| K13 | VSS_17 |
| K14 | VSS_18 |
| K15 | VSS_19 |
| K16 | VSS_20 |
| K17 | VSS_21 |
| K18 | VSS_22 |
| K19 | VSS_23 |
| K20 | VDD_AL_22 |
| K21 | VDD_AL_23 |
| K22 | VDD_AL_24 |
| K23 | Ref_rext_0 |
| K24 | VSS_24 |
| K25 | P1_D3N |
| K26 | P1_D3P |
| L1 | Reserved_75 |
| L2 | Reserved_74 |
| L3 | VSS_25 |
| L4 | Ref_filt_2 |
| L5 | VSS_26 |
| L6 | VDD_25 |
| L7 | VDD_26 |
| L8 | VSS_27 |
| L9 | VSS_28 |
| L10 | VSS_29 |
| L11 | VSS_30 |
| L12 | VSS_31 |
| L13 | VSS_32 |
| L14 | VSS_33 |
| L15 | VSS_34 |
| L16 | VSS_35 |
| L17 | VSS_36 |
| L18 | VSS_37 |
| L19 | VSS_38 |
| L20 | VDD_27 |
| L21 | VDD_28 |
| L22 | VSS_39 |
| L23 | Ref_filt_0 |
| L24 | VSS_40 |
| L25 | P0_D0N |
| L26 | P0_D0P |
| M1 | Reserved_77 |
| M2 | Reserved_76 |
| M3 | VDD_AH_31 |
| M4 | VDD_AH_32 |
| M5 | VDD_AH_33 |
| M6 | VDD_29 |
| M7 | VDD_30 |
| M8 | VSS_41 |
| M9 | VSS_42 |
| M10 | VSS_43 |
| M11 | VSS_44 |
| M12 | VSS_45 |
| M13 | VSS_46 |
| M14 | VSS_47 |
| M15 | VSS_48 |
| M16 | VSS_49 |
| M17 | VSS_50 |
| M18 | VSS_51 |
| M19 | VSS_52 |
| M20 | VDD_31 |
| M21 | VDD_32 |
| M22 | VDD_AH_34 |
| M23 | VDD_AH_35 |

Pins by number *(continued)*

| | |
|---|---|
| M24 | VDD_AH_36 |
| M25 | P0_D1N |
| M26 | P0_D1P |
| N1 | Reserved_79 |
| N2 | Reserved_78 |
| N3 | VSS_53 |
| N4 | VSS_54 |
| N5 | VSS_55 |
| N6 | VDD_33 |
| N7 | VDD_34 |
| N8 | VSS_56 |
| N9 | VSS_57 |
| N10 | VSS_58 |
| N11 | VSS_59 |
| N12 | VSS_60 |
| N13 | VSS_61 |
| N14 | VSS_62 |
| N15 | VSS_63 |
| N16 | VSS_64 |
| N17 | VSS_65 |
| N18 | VSS_66 |
| N19 | VSS_67 |
| N20 | VDD_35 |
| N21 | VDD_36 |
| N22 | VSS_68 |
| N23 | VSS_69 |
| N24 | VSS_70 |
| N25 | P0_D2N |
| N26 | P0_D2P |
| P1 | Reserved_81 |
| P2 | Reserved_80 |
| P3 | VSS_71 |
| P4 | Reserved_24 |
| P5 | VDD_IO_7 |
| P6 | VDD_37 |
| P7 | VDD_38 |
| P8 | VSS_72 |
| P9 | VSS_73 |
| P10 | VSS_74 |

| | |
|---|---|
| P11 | VSS_75 |
| P12 | VSS_76 |
| P13 | VSS_77 |
| P14 | VSS_78 |
| P15 | VSS_79 |
| P16 | VSS_80 |
| P17 | VSS_81 |
| P18 | VSS_82 |
| P19 | VSS_83 |
| P20 | VDD_39 |
| P21 | VDD_40 |
| P22 | VDD_IODDR_1 |
| P23 | VSS_84 |
| P24 | VSS_85 |
| P25 | P0_D3N |
| P26 | P0_D3P |
| R1 | GPIO_31 |
| R2 | GPIO_30 |
| R3 | GPIO_29 |
| R4 | GPIO_28 |
| R5 | VDD_IO_8 |
| R6 | VDD_41 |
| R7 | VDD_42 |
| R8 | VSS_86 |
| R9 | VSS_87 |
| R10 | VSS_88 |
| R11 | VSS_89 |
| R12 | VSS_90 |
| R13 | VSS_91 |
| R14 | VSS_92 |
| R15 | VSS_93 |
| R16 | VSS_94 |
| R17 | VSS_95 |
| R18 | VSS_96 |
| R19 | VSS_97 |
| R20 | VDD_43 |
| R21 | VDD_44 |
| R22 | VDD_IODDR_2 |
| R23 | Reserved_20 |

| | |
|---|---|
| R24 | Reserved_19 |
| R25 | DDR_Rext |
| R26 | DDR_Vref |
| T1 | GPIO_27 |
| T2 | GPIO_26 |
| T3 | GPIO_25 |
| T4 | GPIO_24 |
| T5 | VDD_IO_9 |
| T6 | VDD_45 |
| T7 | VDD_46 |
| T8 | VSS_98 |
| T9 | VSS_99 |
| T10 | VSS_100 |
| T11 | VSS_101 |
| T12 | VSS_102 |
| T13 | VSS_103 |
| T14 | VSS_104 |
| T15 | VSS_105 |
| T16 | VSS_106 |
| T17 | VSS_107 |
| T18 | VSS_108 |
| T19 | VSS_109 |
| T20 | VDD_47 |
| T21 | VDD_48 |
| T22 | VDD_IODDR_3 |
| T23 | Reserved_21 |
| T24 | DDR_A13 |
| T25 | DDR_A12 |
| T26 | DDR_A11 |
| U1 | GPIO_23 |
| U2 | GPIO_22 |
| U3 | GPIO_21 |
| U4 | GPIO_20 |
| U5 | VDD_IO_10 |
| U6 | VSS_110 |
| U7 | VSS_111 |
| U8 | VSS_112 |
| U9 | VSS_113 |
| U10 | VSS_114 |

Pins by number *(continued)*

| | |
|---|---|
| U11 | VSS_115 |
| U12 | VSS_116 |
| U13 | VSS_117 |
| U14 | VSS_118 |
| U15 | VSS_119 |
| U16 | VSS_120 |
| U17 | VSS_121 |
| U18 | VSS_122 |
| U19 | VSS_123 |
| U20 | VSS_124 |
| U21 | VSS_125 |
| U22 | VDD_IODDR_4 |
| U23 | DDR_A7 |
| U24 | DDR_A9 |
| U25 | DDR_A6 |
| U26 | DDR_A8 |
| V1 | GPIO_19 |
| V2 | GPIO_18 |
| V3 | GPIO_17 |
| V4 | GPIO_16 |
| V5 | VDD_IO_11 |
| V6 | VDD_49 |
| V7 | VDD_50 |
| V8 | VDD_51 |
| V9 | VDD_52 |
| V10 | VDD_53 |
| V11 | VDD_54 |
| V12 | VDD_55 |
| V13 | VDD_56 |
| V14 | VDD_57 |
| V15 | VDD_58 |
| V16 | VDD_59 |
| V17 | VDD_60 |
| V18 | VDD_61 |
| V19 | VDD_62 |
| V20 | VDD_63 |
| V21 | VDD_64 |
| V22 | VDD_IODDR_5 |
| V23 | DDR_A3 |
| V24 | DDR_A5 |
| V25 | DDR_A2 |
| V26 | DDR_A4 |
| W1 | GPIO_15 |
| W2 | GPIO_14 |
| W3 | GPIO_13 |
| W4 | GPIO_12 |
| W5 | VDD_IO_12 |
| W6 | VDD_65 |
| W7 | VDD_66 |
| W8 | VDD_67 |
| W9 | VDD_68 |
| W10 | VDD_69 |
| W11 | VDD_70 |
| W12 | VDD_71 |
| W13 | VDD_72 |
| W14 | VDD_73 |
| W15 | VDD_74 |
| W16 | VDD_75 |
| W17 | VDD_76 |
| W18 | VDD_77 |
| W19 | VDD_78 |
| W20 | VDD_79 |
| W21 | VDD_80 |
| W22 | VDD_IODDR_6 |
| W23 | DDR_A10 |
| W24 | DDR_A1 |
| W25 | DDR_nCAS |
| W26 | DDR_A0 |
| Y1 | GPIO_11 |
| Y2 | GPIO_10 |
| Y3 | GPIO_9 |
| Y4 | GPIO_8 |
| Y5 | VDD_IO_13 |
| Y6 | Reserved_146 |
| Y7 | Reserved_141 |
| Y8 | RefClk_P |
| Y9 | Reserved_137 |
| Y10 | Reserved_134 |
| Y11 | Reserved_129 |
| Y12 | VSS_126 |
| Y13 | Reserved_126 |
| Y14 | Reserved_121 |
| Y15 | Reserved_118 |
| Y16 | VSS_127 |
| Y17 | Reserved_113 |
| Y18 | SerDes1_TxP |
| Y19 | SerDes0_TxP |
| Y20 | VSS_128 |
| Y21 | Reserved_102 |
| Y22 | VDD_IODDR_7 |
| Y23 | DDR_BA0 |
| Y24 | DDR_BA1 |
| Y25 | DDR_ODT |
| Y26 | DDR_nRAS |

## 9.4 Pins by Name for VSC7424-02

This section provides an alphabetical list of the VSC7424-02 pins.

| | |
|---|---|
| COMA_MODE | C3 |
| DDR_A0 | W26 |
| DDR_A1 | W24 |
| DDR_A2 | V25 |
| DDR_A3 | V23 |
| DDR_A4 | V26 |
| DDR_A5 | V24 |
| DDR_A6 | U25 |
| DDR_A7 | U23 |
| DDR_A8 | U26 |
| DDR_A9 | U24 |
| DDR_A10 | W23 |
| DDR_A11 | T26 |
| DDR_A12 | T25 |
| DDR_A13 | T24 |
| DDR_BA0 | Y23 |
| DDR_BA1 | Y24 |
| DDR_BA2 | AA24 |
| DDR_CK | AA25 |
| DDR_CKE | AB24 |
| DDR_CKn | AA26 |
| DDR_DM | AE25 |
| DDR_DQ0 | AC26 |
| DDR_DQ1 | AC23 |
| DDR_DQ2 | AB25 |
| DDR_DQ3 | AB23 |
| DDR_DQ4 | AC24 |
| DDR_DQ5 | AB26 |
| DDR_DQ6 | AD24 |
| DDR_DQ7 | AC25 |
| DDR_DQS | AD25 |
| DDR_DQSn | AD26 |
| DDR_nCAS | W25 |
| DDR_nRAS | Y26 |
| DDR_nWE | AA23 |
| DDR_ODT | Y25 |
| DDR_Rext | R25 |
| DDR_Vref | R26 |
| GPIO_0 | AB4 |
| GPIO_1 | AB3 |
| GPIO_2 | AB2 |
| GPIO_3 | AB1 |
| GPIO_4 | AA4 |
| GPIO_5 | AA3 |
| GPIO_6 | AA2 |
| GPIO_7 | AA1 |
| GPIO_8 | Y4 |
| GPIO_9 | Y3 |
| GPIO_10 | Y2 |
| GPIO_11 | Y1 |
| GPIO_12 | W4 |
| GPIO_13 | W3 |
| GPIO_14 | W2 |
| GPIO_15 | W1 |
| GPIO_16 | V4 |
| GPIO_17 | V3 |
| GPIO_18 | V2 |
| GPIO_19 | V1 |
| GPIO_20 | U4 |
| GPIO_21 | U3 |
| GPIO_22 | U2 |
| GPIO_23 | U1 |
| GPIO_24 | T4 |
| GPIO_25 | T3 |
| GPIO_26 | T2 |
| GPIO_27 | T1 |
| GPIO_28 | R4 |
| GPIO_29 | R3 |
| GPIO_30 | R2 |
| GPIO_31 | R1 |
| JTAG_CLK | D17 |
| JTAG_DI | D18 |
| JTAG_DO | D19 |
| JTAG_TMS | D20 |
| JTAG_TRST | D21 |
| MDC | AF4 |
| MDIO | AF3 |
| nRESET | C4 |
| P0_D0N | L25 |
| P0_D0P | L26 |
| P0_D1N | M25 |
| P0_D1P | M26 |
| P0_D2N | N25 |
| P0_D2P | N26 |
| P0_D3N | P25 |
| P0_D3P | P26 |
| P1_D0N | G25 |
| P1_D0P | G26 |
| P1_D1N | H25 |
| P1_D1P | H26 |
| P1_D2N | J25 |
| P1_D2P | J26 |
| P1_D3N | K25 |
| P1_D3P | K26 |
| P2_D0N | C25 |
| P2_D0P | C26 |
| P2_D1N | D25 |
| P2_D1P | D26 |
| P2_D2N | E25 |
| P2_D2P | E26 |
| P2_D3N | F25 |
| P2_D3P | F26 |
| P3_D0N | B22 |
| P3_D0P | A22 |
| P3_D1N | B23 |
| P3_D1P | A23 |
| P3_D2N | B24 |
| P3_D2P | A24 |
| P3_D3N | B25 |
| P3_D3P | A25 |

Pins by name *(continued)*

| | |
|---|---|
| P4_D0N | B18 |
| P4_D0P | A18 |
| P4_D1N | B19 |
| P4_D1P | A19 |
| P4_D2N | B20 |
| P4_D2P | A20 |
| P4_D3N | B21 |
| P4_D3P | A21 |
| P5_D0N | B14 |
| P5_D0P | A14 |
| P5_D1N | B15 |
| P5_D1P | A15 |
| P5_D2N | B16 |
| P5_D2P | A16 |
| P5_D3N | B17 |
| P5_D3P | A17 |
| P6_D0N | B10 |
| P6_D0P | A10 |
| P6_D1N | B11 |
| P6_D1P | A11 |
| P6_D2N | B12 |
| P6_D2P | A12 |
| P6_D3N | B13 |
| P6_D3P | A13 |
| P7_D0N | B6 |
| P7_D0P | A6 |
| P7_D1N | B7 |
| P7_D1P | A7 |
| P7_D2N | B8 |
| P7_D2P | A8 |
| P7_D3N | B9 |
| P7_D3P | A9 |
| Ref_filt_0 | L23 |
| Ref_filt_1 | E14 |
| Ref_filt_2 | L4 |
| Ref_rext_0 | K23 |
| Ref_rext_1 | E13 |
| Ref_rext_2 | K4 |
| RefClk_N | AA8 |

| | |
|---|---|
| RefClk_P | Y8 |
| RefClk_Sel0 | C12 |
| RefClk_Sel1 | C13 |
| RefClk_Sel2 | C14 |
| Reserved_1 | C6 |
| Reserved_4 | C11 |
| Reserved_5 | C18 |
| Reserved_6 | C17 |
| Reserved_7 | C16 |
| Reserved_8 | C15 |
| Reserved_10 | G23 |
| Reserved_11 | H23 |
| Reserved_12 | D14 |
| Reserved_13 | D13 |
| Reserved_14 | H4 |
| Reserved_15 | G4 |
| Reserved_17 | AE2 |
| Reserved_18 | AD3 |
| Reserved_19 | R24 |
| Reserved_20 | R23 |
| Reserved_21 | T23 |
| Reserved_22 | AE8 |
| Reserved_23 | AF8 |
| Reserved_24 | P4 |
| Reserved_50 | B5 |
| Reserved_51 | A5 |
| Reserved_52 | B4 |
| Reserved_53 | A4 |
| Reserved_54 | B3 |
| Reserved_55 | A3 |
| Reserved_56 | B2 |
| Reserved_57 | A2 |
| Reserved_58 | C2 |
| Reserved_59 | C1 |
| Reserved_60 | D2 |
| Reserved_61 | D1 |
| Reserved_62 | E2 |
| Reserved_63 | E1 |
| Reserved_64 | F2 |

| | |
|---|---|
| Reserved_65 | F1 |
| Reserved_66 | G2 |
| Reserved_67 | G1 |
| Reserved_68 | H2 |
| Reserved_69 | H1 |
| Reserved_70 | J2 |
| Reserved_71 | J1 |
| Reserved_72 | K2 |
| Reserved_73 | K1 |
| Reserved_74 | L2 |
| Reserved_75 | L1 |
| Reserved_76 | M2 |
| Reserved_77 | M1 |
| Reserved_78 | N2 |
| Reserved_79 | N1 |
| Reserved_80 | P2 |
| Reserved_81 | P1 |
| Reserved_100 | AE21 |
| Reserved_101 | AF21 |
| Reserved_102 | Y21 |
| Reserved_103 | AA21 |
| Reserved_112 | AA17 |
| Reserved_113 | Y17 |
| Reserved_114 | AF17 |
| Reserved_115 | AE17 |
| Reserved_116 | AE15 |
| Reserved_117 | AF15 |
| Reserved_118 | Y15 |
| Reserved_119 | AA15 |
| Reserved_120 | AA14 |
| Reserved_121 | Y14 |
| Reserved_122 | AF14 |
| Reserved_123 | AE14 |
| Reserved_124 | AE13 |
| Reserved_125 | AF13 |
| Reserved_126 | Y13 |
| Reserved_127 | AA13 |
| Reserved_128 | AA11 |
| Reserved_129 | Y11 |

Pins by name *(continued)*

| | |
|---|---|
| Reserved_130 | AF11 |
| Reserved_131 | AE11 |
| Reserved_132 | AE10 |
| Reserved_133 | AF10 |
| Reserved_134 | Y10 |
| Reserved_135 | AA10 |
| Reserved_136 | AA9 |
| Reserved_137 | Y9 |
| Reserved_138 | AF9 |
| Reserved_139 | AE9 |
| Reserved_140 | AA7 |
| Reserved_141 | Y7 |
| Reserved_142 | AF7 |
| Reserved_143 | AE7 |
| Reserved_144 | AE6 |
| Reserved_145 | AF6 |
| Reserved_146 | Y6 |
| Reserved_147 | AA6 |
| Reserved_201 | C19 |
| Reserved_202 | C20 |
| Reserved_203 | C21 |
| Reserved_204 | C24 |
| Reserved_205 | D3 |
| Reserved_206 | D6 |
| Reserved_207 | D7 |
| Reserved_208 | D8 |
| Reserved_209 | D9 |
| Reserved_211 | D12 |
| Reserved_212 | D15 |
| Reserved_213 | D22 |
| Reserved_214 | D23 |
| Reserved_215 | D24 |
| Reserved_216 | E3 |
| Reserved_217 | E24 |
| Reserved_218 | F3 |
| Reserved_219 | F13 |
| Reserved_220 | F14 |
| Reserved_221 | F24 |
| Reserved_223 | G14 |
| Reserved_225 | H14 |
| Reserved_232 | J14 |
| Reserved_233 | J15 |
| Reserved_234 | J16 |
| Reserved_235 | J17 |
| Reserved_236 | J18 |
| Reserved_237 | J19 |
| Reserved_240 | J8 |
| Reserved_241 | J9 |
| Reserved_242 | J10 |
| Reserved_243 | J11 |
| Reserved_244 | J12 |
| Reserved_245 | J13 |
| Reserved_246 | H13 |
| Reserved_247 | G13 |
| Reserved_248 | D10 |
| SerDes_Rext_0 | AE22 |
| SerDes_Rext_1 | AF22 |
| SerDes0_RxN | AF19 |
| SerDes0_RxP | AE19 |
| SerDes0_TxN | AA19 |
| SerDes0_TxP | Y19 |
| SerDes1_RxN | AF18 |
| SerDes1_RxP | AE18 |
| SerDes1_TxN | AA18 |
| SerDes1_TxP | Y18 |
| SI_Clk | AD1 |
| SI_DI | AD2 |
| SI_DO | AC1 |
| SI_nEn | AC2 |
| THERMDA | C23 |
| THERMDC_VSS | C22 |
| VCORE_CFG0 | C7 |
| VCORE_CFG1 | C8 |
| VCORE_CFG2 | C9 |
| VCore_ICE_nEn | C10 |
| VDD_1 | G6 |
| VDD_2 | G7 |
| VDD_3 | G8 |
| VDD_4 | G11 |
| VDD_5 | G12 |
| VDD_6 | G15 |
| VDD_7 | G16 |
| VDD_8 | G19 |
| VDD_9 | G20 |
| VDD_10 | G21 |
| VDD_11 | H6 |
| VDD_12 | H7 |
| VDD_13 | H8 |
| VDD_14 | H9 |
| VDD_15 | H10 |
| VDD_16 | H11 |
| VDD_17 | H12 |
| VDD_18 | H15 |
| VDD_19 | H16 |
| VDD_20 | H17 |
| VDD_21 | H18 |
| VDD_22 | H19 |
| VDD_23 | H20 |
| VDD_24 | H21 |
| VDD_25 | L6 |
| VDD_26 | L7 |
| VDD_27 | L20 |
| VDD_28 | L21 |
| VDD_29 | M6 |
| VDD_30 | M7 |
| VDD_31 | M20 |
| VDD_32 | M21 |
| VDD_33 | N6 |
| VDD_34 | N7 |
| VDD_35 | N20 |
| VDD_36 | N21 |
| VDD_37 | P6 |
| VDD_38 | P7 |
| VDD_39 | P20 |
| VDD_40 | P21 |
| VDD_41 | R6 |
| VDD_42 | R7 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_43 | R20 |
| VDD_44 | R21 |
| VDD_45 | T6 |
| VDD_46 | T7 |
| VDD_47 | T20 |
| VDD_48 | T21 |
| VDD_49 | V6 |
| VDD_50 | V7 |
| VDD_51 | V8 |
| VDD_52 | V9 |
| VDD_53 | V10 |
| VDD_54 | V11 |
| VDD_55 | V12 |
| VDD_56 | V13 |
| VDD_57 | V14 |
| VDD_58 | V15 |
| VDD_59 | V16 |
| VDD_60 | V17 |
| VDD_61 | V18 |
| VDD_62 | V19 |
| VDD_63 | V20 |
| VDD_64 | V21 |
| VDD_65 | W6 |
| VDD_66 | W7 |
| VDD_67 | W8 |
| VDD_68 | W9 |
| VDD_69 | W10 |
| VDD_70 | W11 |
| VDD_71 | W12 |
| VDD_72 | W13 |
| VDD_73 | W14 |
| VDD_74 | W15 |
| VDD_75 | W16 |
| VDD_76 | W17 |
| VDD_77 | W18 |
| VDD_78 | W19 |
| VDD_79 | W20 |
| VDD_80 | W21 |
| VDD_A_1 | AC6 |
| VDD_A_2 | AC7 |
| VDD_A_3 | AC8 |
| VDD_A_4 | AC9 |
| VDD_A_5 | AC10 |
| VDD_A_6 | AC11 |
| VDD_A_7 | AC12 |
| VDD_A_8 | AC13 |
| VDD_A_9 | AC14 |
| VDD_A_10 | AC15 |
| VDD_A_11 | AC16 |
| VDD_A_12 | AC17 |
| VDD_A_13 | AC18 |
| VDD_A_14 | AC19 |
| VDD_A_15 | AC20 |
| VDD_A_16 | AC21 |
| VDD_AH_1 | D4 |
| VDD_AH_2 | D5 |
| VDD_AH_3 | F7 |
| VDD_AH_4 | D11 |
| VDD_AH_5 | D16 |
| VDD_AH_6 | F20 |
| VDD_AH_7 | E4 |
| VDD_AH_8 | E5 |
| VDD_AH_9 | E8 |
| VDD_AH_10 | E11 |
| VDD_AH_11 | E12 |
| VDD_AH_12 | E15 |
| VDD_AH_13 | E16 |
| VDD_AH_14 | E19 |
| VDD_AH_15 | E22 |
| VDD_AH_16 | E23 |
| VDD_AH_17 | F4 |
| VDD_AH_18 | F5 |
| VDD_AH_19 | F8 |
| VDD_AH_20 | F11 |
| VDD_AH_21 | F12 |
| VDD_AH_22 | F15 |
| VDD_AH_23 | F16 |
| VDD_AH_24 | F19 |
| VDD_AH_25 | F22 |
| VDD_AH_26 | F23 |
| VDD_AH_27 | J3 |
| VDD_AH_28 | J4 |
| VDD_AH_29 | J23 |
| VDD_AH_30 | J24 |
| VDD_AH_31 | M3 |
| VDD_AH_32 | M4 |
| VDD_AH_33 | M5 |
| VDD_AH_34 | M22 |
| VDD_AH_35 | M23 |
| VDD_AH_36 | M24 |
| VDD_AL_1 | E9 |
| VDD_AL_2 | E10 |
| VDD_AL_3 | E17 |
| VDD_AL_4 | E18 |
| VDD_AL_5 | F9 |
| VDD_AL_6 | F10 |
| VDD_AL_7 | F17 |
| VDD_AL_8 | F18 |
| VDD_AL_9 | G9 |
| VDD_AL_10 | G10 |
| VDD_AL_11 | G17 |
| VDD_AL_12 | G18 |
| VDD_AL_13 | J5 |
| VDD_AL_14 | J6 |
| VDD_AL_15 | J7 |
| VDD_AL_16 | J20 |
| VDD_AL_17 | J21 |
| VDD_AL_18 | J22 |
| VDD_AL_19 | K5 |
| VDD_AL_20 | K6 |
| VDD_AL_21 | K7 |
| VDD_AL_22 | K20 |
| VDD_AL_23 | K21 |
| VDD_AL_24 | K22 |
| VDD_IO_1 | E6 |
| VDD_IO_2 | E7 |
| VDD_IO_3 | E20 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_IO_4 | E21 |
| VDD_IO_5 | F6 |
| VDD_IO_6 | F21 |
| VDD_IO_7 | P5 |
| VDD_IO_8 | R5 |
| VDD_IO_9 | T5 |
| VDD_IO_10 | U5 |
| VDD_IO_11 | V5 |
| VDD_IO_12 | W5 |
| VDD_IO_13 | Y5 |
| VDD_IO_14 | AA5 |
| VDD_IO_15 | AB5 |
| VDD_IO_16 | AC4 |
| VDD_IO_17 | AC5 |
| VDD_IO_18 | AD4 |
| VDD_IO_19 | AE3 |
| VDD_IO_20 | AF2 |
| VDD_IO_21 | C5 |
| VDD_IODDR_1 | P22 |
| VDD_IODDR_2 | R22 |
| VDD_IODDR_3 | T22 |
| VDD_IODDR_4 | U22 |
| VDD_IODDR_5 | V22 |
| VDD_IODDR_6 | W22 |
| VDD_IODDR_7 | Y22 |
| VDD_IODDR_8 | AA22 |
| VDD_IODDR_9 | AB22 |
| VDD_IODDR_10 | AC22 |
| VDD_IODDR_11 | AD23 |
| VDD_IODDR_12 | AE24 |
| VDD_IODDR_13 | AF25 |
| VDD_IODDR_14 | AF24 |
| VDD_VS_1 | AD6 |
| VDD_VS_2 | AD7 |
| VDD_VS_3 | AD8 |
| VDD_VS_4 | AD9 |
| VDD_VS_5 | AD10 |
| VDD_VS_6 | AD11 |
| VDD_VS_7 | AD12 |
| VDD_VS_8 | AD13 |
| VDD_VS_9 | AD14 |
| VDD_VS_10 | AD15 |
| VDD_VS_11 | AD16 |
| VDD_VS_12 | AD17 |
| VDD_VS_13 | AD18 |
| VDD_VS_14 | AD19 |
| VDD_VS_15 | AD20 |
| VDD_VS_16 | AD21 |
| VSS_1 | B1 |
| VSS_2 | B26 |
| VSS_3 | G3 |
| VSS_4 | G5 |
| VSS_5 | G22 |
| VSS_6 | G24 |
| VSS_7 | H3 |
| VSS_8 | H5 |
| VSS_9 | H22 |
| VSS_10 | H24 |
| VSS_11 | K3 |
| VSS_12 | K8 |
| VSS_13 | K9 |
| VSS_14 | K10 |
| VSS_15 | K11 |
| VSS_16 | K12 |
| VSS_17 | K13 |
| VSS_18 | K14 |
| VSS_19 | K15 |
| VSS_20 | K16 |
| VSS_21 | K17 |
| VSS_22 | K18 |
| VSS_23 | K19 |
| VSS_24 | K24 |
| VSS_25 | L3 |
| VSS_26 | L5 |
| VSS_27 | L8 |
| VSS_28 | L9 |
| VSS_29 | L10 |
| VSS_30 | L11 |
| VSS_31 | L12 |
| VSS_32 | L13 |
| VSS_33 | L14 |
| VSS_34 | L15 |
| VSS_35 | L16 |
| VSS_36 | L17 |
| VSS_37 | L18 |
| VSS_38 | L19 |
| VSS_39 | L22 |
| VSS_40 | L24 |
| VSS_41 | M8 |
| VSS_42 | M9 |
| VSS_43 | M10 |
| VSS_44 | M11 |
| VSS_45 | M12 |
| VSS_46 | M13 |
| VSS_47 | M14 |
| VSS_48 | M15 |
| VSS_49 | M16 |
| VSS_50 | M17 |
| VSS_51 | M18 |
| VSS_52 | M19 |
| VSS_53 | N3 |
| VSS_54 | N4 |
| VSS_55 | N5 |
| VSS_56 | N8 |
| VSS_57 | N9 |
| VSS_58 | N10 |
| VSS_59 | N11 |
| VSS_60 | N12 |
| VSS_61 | N13 |
| VSS_62 | N14 |
| VSS_63 | N15 |
| VSS_64 | N16 |
| VSS_65 | N17 |
| VSS_66 | N18 |
| VSS_67 | N19 |
| VSS_68 | N22 |
| VSS_69 | N23 |

Pins by name *(continued)*

| | |
|---|---|
| VSS_70 | N24 |
| VSS_71 | P3 |
| VSS_72 | P8 |
| VSS_73 | P9 |
| VSS_74 | P10 |
| VSS_75 | P11 |
| VSS_76 | P12 |
| VSS_77 | P13 |
| VSS_78 | P14 |
| VSS_79 | P15 |
| VSS_80 | P16 |
| VSS_81 | P17 |
| VSS_82 | P18 |
| VSS_83 | P19 |
| VSS_84 | P23 |
| VSS_85 | P24 |
| VSS_86 | R8 |
| VSS_87 | R9 |
| VSS_88 | R10 |
| VSS_89 | R11 |
| VSS_90 | R12 |
| VSS_91 | R13 |
| VSS_92 | R14 |
| VSS_93 | R15 |
| VSS_94 | R16 |
| VSS_95 | R17 |
| VSS_96 | R18 |
| VSS_97 | R19 |
| VSS_98 | T8 |
| VSS_99 | T9 |
| VSS_100 | T10 |
| VSS_101 | T11 |
| VSS_102 | T12 |
| VSS_103 | T13 |
| VSS_104 | T14 |
| VSS_105 | T15 |
| VSS_106 | T16 |
| VSS_107 | T17 |
| VSS_108 | T18 |
| VSS_109 | T19 |
| VSS_110 | U6 |
| VSS_111 | U7 |
| VSS_112 | U8 |
| VSS_113 | U9 |
| VSS_114 | U10 |
| VSS_115 | U11 |
| VSS_116 | U12 |
| VSS_117 | U13 |
| VSS_118 | U14 |
| VSS_119 | U15 |
| VSS_120 | U16 |
| VSS_121 | U17 |
| VSS_122 | U18 |
| VSS_123 | U19 |
| VSS_124 | U20 |
| VSS_125 | U21 |
| VSS_126 | Y12 |
| VSS_127 | Y16 |
| VSS_128 | Y20 |
| VSS_129 | AB6 |
| VSS_130 | AB7 |
| VSS_131 | AB8 |
| VSS_132 | AB9 |
| VSS_133 | AB10 |
| VSS_134 | AB11 |
| VSS_135 | AB12 |
| VSS_136 | AB13 |
| VSS_137 | AB14 |
| VSS_138 | AB15 |
| VSS_139 | AB16 |
| VSS_140 | AB17 |
| VSS_141 | AB18 |
| VSS_142 | AB19 |
| VSS_143 | AB20 |
| VSS_144 | AB21 |
| VSS_145 | AA12 |
| VSS_146 | AA16 |
| VSS_147 | AA20 |
| VSS_148 | AC3 |
| VSS_149 | AD5 |
| VSS_150 | AD22 |
| VSS_151 | AE1 |
| VSS_152 | AE5 |
| VSS_153 | AE12 |
| VSS_154 | AE16 |
| VSS_155 | AE20 |
| VSS_156 | AE23 |
| VSS_157 | AE26 |
| VSS_158 | AF5 |
| VSS_159 | AF12 |
| VSS_160 | AF16 |
| VSS_161 | AF20 |
| VSS_162 | AF23 |
| VSS_163 | AE4 |

# 10 Pin Descriptions for VSC7425-02

The VSC7425-02 device has 672 pins, which are described in this section.

The pin information is also provided as an attached Microsoft Excel file, so that you can copy it electronically. In Adobe Reader, double-click the attachment icon.

## 10.1 Pin Diagram for VSC7425-02

The following illustration shows the pin diagram for the VSC7425-02 device. For clarity, the device is shown in two halves, the top left and top right.

***Figure 106*** • **Pin Diagram for VSC7425-02, Top Left**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | P8_D0P | P8_D1P | P8_D2P | P8_D3P | P7_D0P | P7_D1P | P7_D2P | P7_D3P | P6_D0P | P6_D1P | P6_D2P | P6_D3P |
| B | VSS_1 | P8_D0N | P8_D1N | P8_D2N | P8_D3N | P7_D0N | P7_D1N | P7_D2N | P7_D3N | P6_D0N | P6_D1N | P6_D2N | P6_D3N |
| C | P9_D3P | P9_D3N | COMA_MODE | nRESET | VDD_IO_21 | Reserved_1 | VCORE_CFG0 | VCORE_CFG1 | VCORE_CFG2 | VCore_ICE_nEn | Reserved_4 | RefClk_Sel0 | RefClk_Sel1 |
| D | P9_D2P | P9_D2N | Reserved_205 | VDD_AH_1 | VDD_AH_2 | Reserved_206 | Reserved_207 | Reserved_208 | Reserved_209 | Reserved_248 | VDD_AH_4 | Reserved_211 | Reserved_13 |
| E | P9_D1P | P9_D1N | Reserved_216 | VDD_AH_7 | VDD_AH_8 | VDD_IO_1 | VDD_IO_2 | VDD_AH_9 | VDD_AL_1 | VDD_AL_2 | VDD_AH_10 | VDD_AH_11 | Ref_rext_1 |
| F | P9_D0P | P9_D0N | Reserved_218 | VDD_AH_17 | VDD_AH_18 | VDD_IO_5 | VDD_AH_3 | VDD_AH_19 | VDD_AL_5 | VDD_AL_6 | VDD_AH_20 | VDD_AH_21 | Reserved_219 |
| G | P10_D3P | P10_D3N | VSS_3 | Reserved_15 | VSS_4 | VDD_1 | VDD_2 | VDD_3 | VDD_AL_9 | VDD_AL_10 | VDD_4 | VDD_5 | Reserved_247 |
| H | P10_D2P | P10_D2N | VSS_7 | Reserved_14 | VSS_8 | VDD_11 | VDD_12 | VDD_13 | VDD_14 | VDD_15 | VDD_16 | VDD_17 | Reserved_246 |
| J | P10_D1P | P10_D1N | VDD_AH_27 | VDD_AH_28 | VDD_AL_13 | VDD_AL_14 | VDD_AL_15 | Reserved_240 | Reserved_241 | Reserved_242 | Reserved_243 | Reserved_244 | Reserved_245 |
| K | P10_D0P | P10_D0N | VSS_11 | Ref_rext_2 | VDD_AL_19 | VDD_AL_20 | VDD_AL_21 | VSS_12 | VSS_13 | VSS_14 | VSS_15 | VSS_16 | VSS_17 |
| L | P11_D3P | P11_D3N | VSS_25 | Ref_filt_2 | VSS_26 | VDD_25 | VDD_26 | VSS_27 | VSS_28 | VSS_29 | VSS_30 | VSS_31 | VSS_32 |
| M | P11_D2P | P11_D2N | VDD_AH_31 | VDD_AH_32 | VDD_AH_33 | VDD_29 | VDD_30 | VSS_41 | VSS_42 | VSS_43 | VSS_44 | VSS_45 | VSS_46 |
| N | P11_D1P | P11_D1N | VSS_53 | VSS_54 | VSS_55 | VDD_33 | VDD_34 | VSS_56 | VSS_57 | VSS_58 | VSS_59 | VSS_60 | VSS_61 |
| P | P11_D0P | P11_D0N | VSS_71 | Reserved_24 | VDD_IO_7 | VDD_37 | VDD_38 | VSS_72 | VSS_73 | VSS_74 | VSS_75 | VSS_76 | VSS_77 |
| R | GPIO_31 | GPIO_30 | GPIO_29 | GPIO_28 | VDD_IO_8 | VDD_41 | VDD_42 | VSS_86 | VSS_87 | VSS_88 | VSS_89 | VSS_90 | VSS_91 |
| T | GPIO_27 | GPIO_26 | GPIO_25 | GPIO_24 | VDD_IO_9 | VDD_45 | VDD_46 | VSS_98 | VSS_99 | VSS_100 | VSS_101 | VSS_102 | VSS_103 |
| U | GPIO_23 | GPIO_22 | GPIO_21 | GPIO_20 | VDD_IO_10 | VSS_110 | VSS_111 | VSS_112 | VSS_113 | VSS_114 | VSS_115 | VSS_116 | VSS_117 |
| V | GPIO_19 | GPIO_18 | GPIO_17 | GPIO_16 | VDD_IO_11 | VDD_49 | VDD_50 | VDD_51 | VDD_52 | VDD_53 | VDD_54 | VDD_55 | VDD_56 |
| W | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | VDD_IO_12 | VDD_65 | VDD_66 | VDD_67 | VDD_68 | VDD_69 | VDD_70 | VDD_71 | VDD_72 |
| Y | GPIO_11 | GPIO_10 | GPIO_9 | GPIO_8 | VDD_IO_13 | Reserved_146 | Reserved_141 | RefClk_P | Reserved_137 | SerDes5_TxP | SerDes4_TxP | VSS_126 | Reserved_126 |
| AA | GPIO_7 | GPIO_6 | GPIO_5 | GPIO_4 | VDD_IO_14 | Reserved_147 | Reserved_140 | RefClk_N | Reserved_136 | SerDes5_TxN | SerDes4_TxN | VSS_145 | Reserved_127 |
| AB | GPIO_3 | GPIO_2 | GPIO_1 | GPIO_0 | VDD_IO_15 | VSS_129 | VSS_130 | VSS_131 | VSS_132 | VSS_133 | VSS_134 | VSS_135 | VSS_136 |
| AC | SI_DO | SI_nEn | VSS_148 | VDD_IO_16 | VDD_IO_17 | VDD_A_1 | VDD_A_2 | VDD_A_3 | VDD_A_4 | VDD_A_5 | VDD_A_6 | VDD_A_7 | VDD_A_8 |
| AD | SI_Clk | SI_DI | Reserved_18 | VDD_IO_18 | VSS_149 | VDD_VS_1 | VDD_VS_2 | VDD_VS_3 | VDD_VS_4 | VDD_VS_5 | VDD_VS_6 | VDD_VS_7 | VDD_VS_8 |
| AE | VSS_151 | Reserved_17 | VDD_IO_19 | VSS_163 | VSS_152 | Reserved_144 | Reserved_143 | Reserved_22 | Reserved_139 | SerDes5_RxP | SerDes4_RxP | VSS_153 | Reserved_124 |
| AF | | VDD_IO_20 | MDIO | MDC | VSS_158 | Reserved_145 | Reserved_142 | Reserved_23 | Reserved_138 | SerDes5_RxN | SerDes4_RxN | VSS_159 | Reserved_125 |

***Figure 107*** • **Pin Diagram for VSC7425-02, Top Right**

| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P5_D0P | P5_D1P | P5_D2P | P5_D3P | P4_D0P | P4_D1P | P4_D2P | P4_D3P | P3_D0P | P3_D1P | P3_D2P | P3_D3P | | A |
| P5_D0N | P5_D1N | P5_D2N | P5_D3N | P4_D0N | P4_D1N | P4_D2N | P4_D3N | P3_D0N | P3_D1N | P3_D2N | P3_D3N | VSS_2 | B |
| RefClk_Sel2 | Reserved_8 | Reserved_7 | Reserved_6 | Reserved_5 | Reserved_201 | Reserved_202 | Reserved_203 | THERMDC_VSS | THERMDA | Reserved_204 | P2_D0N | P2_D0P | C |
| Reserved_12 | Reserved_212 | VDD_AH_5 | JTAG_CLK | JTAG_DI | JTAG_DO | JTAG_TMS | JTAG_TRST | Reserved_213 | Reserved_214 | Reserved_215 | P2_D1N | P2_D1P | D |
| Ref_filt_1 | VDD_AH_12 | VDD_AH_13 | VDD_AL_3 | VDD_AL_4 | VDD_AH_14 | VDD_IO_3 | VDD_IO_4 | VDD_AH_15 | VDD_AH_16 | Reserved_217 | P2_D2N | P2_D2P | E |
| Reserved_220 | VDD_AH_22 | VDD_AH_23 | VDD_AL_7 | VDD_AL_8 | VDD_AH_24 | VDD_AH_6 | VDD_IO_6 | VDD_AH_25 | VDD_AH_26 | Reserved_221 | P2_D3N | P2_D3P | F |
| Reserved_223 | VDD_6 | VDD_7 | VDD_AL_11 | VDD_AL_12 | VDD_8 | VDD_9 | VDD_10 | VSS_5 | Reserved_10 | VSS_6 | P1_D0N | P1_D0P | G |
| Reserved_225 | VDD_18 | VDD_19 | VDD_20 | VDD_21 | VDD_22 | VDD_23 | VDD_24 | VSS_9 | Reserved_11 | VSS_10 | P1_D1N | P1_D1P | H |
| Reserved_232 | Reserved_233 | Reserved_234 | Reserved_235 | Reserved_236 | Reserved_237 | VDD_AL_16 | VDD_AL_17 | VDD_AL_18 | VDD_AH_29 | VDD_AH_30 | P1_D2N | P1_D2P | J |
| VSS_18 | VSS_19 | VSS_20 | VSS_21 | VSS_22 | VSS_23 | VDD_AL_22 | VDD_AL_23 | VDD_AL_24 | Ref_rext_0 | VSS_24 | P1_D3N | P1_D3P | K |
| VSS_33 | VSS_34 | VSS_35 | VSS_36 | VSS_37 | VSS_38 | VDD_27 | VDD_28 | VSS_39 | Ref_filt_0 | VSS_40 | P0_D0N | P0_D0P | L |
| VSS_47 | VSS_48 | VSS_49 | VSS_50 | VSS_51 | VSS_52 | VDD_31 | VDD_32 | VDD_AH_34 | VDD_AH_35 | VDD_AH_36 | P0_D1N | P0_D1P | M |
| VSS_62 | VSS_63 | VSS_64 | VSS_65 | VSS_66 | VSS_67 | VDD_35 | VDD_36 | VSS_68 | VSS_69 | VSS_70 | P0_D2N | P0_D2P | N |
| VSS_78 | VSS_79 | VSS_80 | VSS_81 | VSS_82 | VSS_83 | VDD_39 | VDD_40 | VDD_IODDR_1 | VSS_84 | VSS_85 | P0_D3N | P0_D3P | P |
| VSS_92 | VSS_93 | VSS_94 | VSS_95 | VSS_96 | VSS_97 | VDD_43 | VDD_44 | VDD_IODDR_2 | Reserved_20 | Reserved_19 | DDR_Rext | DDR_Vref | R |
| VSS_104 | VSS_105 | VSS_106 | VSS_107 | VSS_108 | VSS_109 | VDD_47 | VDD_48 | VDD_IODDR_3 | Reserved_21 | DDR_A13 | DDR_A12 | DDR_A11 | T |
| VSS_118 | VSS_119 | VSS_120 | VSS_121 | VSS_122 | VSS_123 | VSS_124 | VSS_125 | VDD_IODDR_4 | DDR_A7 | DDR_A9 | DDR_A6 | DDR_A8 | U |
| VDD_57 | VDD_58 | VDD_59 | VDD_60 | VDD_61 | VDD_62 | VDD_63 | VDD_64 | VDD_IODDR_5 | DDR_A3 | DDR_A5 | DDR_A2 | DDR_A4 | V |
| VDD_73 | VDD_74 | VDD_75 | VDD_76 | VDD_77 | VDD_78 | VDD_79 | VDD_80 | VDD_IODDR_6 | DDR_A10 | DDR_A1 | DDR_nCAS | DDR_A0 | W |
| SerDes3_TxP | SerDes2_TxP | VSS_127 | SerDes_E1_TxP | SerDes1_TxP | SerDes0_TxP | VSS_128 | SerDes_E0_TxP | VDD_IODDR_7 | DDR_BA0 | DDR_BA1 | DDR_ODT | DDR_nRAS | Y |
| SerDes3_TxN | SerDes2_TxN | VSS_146 | SerDes_E1_TxN | SerDes1_TxN | SerDes0_TxN | VSS_147 | SerDes_E0_TxN | VDD_IODDR_8 | DDR_nWE | DDR_BA2 | DDR_CK | DDR_CKn | AA |
| VSS_137 | VSS_138 | VSS_139 | VSS_140 | VSS_141 | VSS_142 | VSS_143 | VSS_144 | VDD_IODDR_9 | DDR_DQ3 | DDR_CKE | DDR_DQ2 | DDR_DQ5 | AB |
| VDD_A_9 | VDD_A_10 | VDD_A_11 | VDD_A_12 | VDD_A_13 | VDD_A_14 | VDD_A_15 | VDD_A_16 | VDD_IODDR_10 | DDR_DQ1 | DDR_DQ4 | DDR_DQ7 | DDR_DQ0 | AC |
| VDD_VS_9 | VDD_VS_10 | VDD_VS_11 | VDD_VS_12 | VDD_VS_13 | VDD_VS_14 | VDD_VS_15 | VDD_VS_16 | VSS_150 | VDD_IODDR_11 | DDR_DQ6 | DDR_DQS | DDR_DQSn | AD |
| SerDes3_RxP | SerDes2_RxP | VSS_154 | SerDes_E1_RxP | SerDes1_RxP | SerDes0_RxP | VSS_155 | SerDes_E0_RxP | SerDes_Rext_0 | VSS_156 | VDD_IODDR_12 | DDR_DM | VSS_157 | AE |
| SerDes3_RxN | SerDes2_RxN | VSS_160 | SerDes_E1_RxN | SerDes1_RxN | SerDes0_RxN | VSS_161 | SerDes_E0_RxN | SerDes_Rext_1 | VSS_162 | VDD_IODDR_14 | VDD_IODDR_13 | | AF |

# 10.2 Pins by Function for VSC7425-02

This section contains the functional pin descriptions for the VSC7425-02 device. The following table lists the definitions for the pin type symbols.

***Table 868*** • **Pin Type Symbol Definitions**

| Symbol | Pin Type | Description |
|---|---|---|
| ABIAS | Analog bias | Analog bias pin. |
| DIFF | Differential | Differential signal pair. |
| I | Input | Input signal. |
| O | Output | Output signal. |
| I/O | Bidirectional | Bidirectional input or output signal. |
| A | Analog input | Analog input for sensing variable voltage levels. |
| PD | Pull-down | On-chip pull-down resistor to VSS. |
| PU | Pull-up | On-chip pull-up resistor to VDD_IO. |
| 3V | | 3.3 V-tolerant. |
| O | Output | Output signal. |

*Table 868 •* **Pin Type Symbol Definitions *(continued)***

| Symbol | Pin Type | Description |
|---|---|---|
| OZ | 3-state output | Output. |
| ST | Schmitt-trigger | Input has Schmitt-trigger circuitry. |
| TD | Termination differential | Internal differential termination. |

## 10.2.1 Analog Bias Signals

The following table lists the pins associated with the analog bias signals.

*Table 869 •* **Analog Bias Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_Rext_[1:0] | A | Analog bias calibration.<br>Connect an external 620 Ω ±1% resistor between SerDes_Rext_1 and SerDes_Rext_0. |
| Ref_filt_[2:0] | A | Reference filter. Connect a 1.0 µF external capacitor between each pin and ground. |
| Ref_rext_[2:0] | A | Reference external resistor. Connect a 2.0 kΩ (1%) resistor between each pin and ground. |

## 10.2.2 Clock Circuits

The following table lists the pins associated with the system clock interface.

*Table 870 •* **System Clock Interface Pins**

| Name | Type | Description |
|---|---|---|
| RefClk_Sel[2:0] | I, PD | Reference clock frequency selection.<br>0: Connect to pull-down or leave floating.<br>1: Connect to pull-up to $V_{DD_IO}$.<br>Coding:<br>000: 125 MHz (default).<br>001: 156.25 MHz.<br>010: Reserved.<br>011: Reserved.<br>100: 25 MHz.<br>101: Reserved.<br>110: Reserved.<br>111: Reserved. |
| RefClk_P<br>RefClk_N | I, Diff | Reference clock input.<br>The input can be either differential or single-ended. In differential mode, REFCLK_P is the true part of the differential signal, and REFCLK_N is the complement part of the differential signal.<br>In single-ended mode, REFCLK_P is used as single-ended LVTTL input, and the REFCLK_N should be pulled to $V_{DD_A}$.<br>Required applied frequency depends on RefClk_Sel[2:0] input state. See description for RefClk_Sel[2:0] pins. |

### 10.2.3 DDR2 SDRAM Interface

The following table lists the pins associated with the DDR2 SDRAM interface.

***Table 871 •*** **DDR2 SDRAM Pins**

| Name | Type | Description |
|---|---|---|
| DDR_CK<br>DDR_CKn | 0, Diff | SDRAM differential clock. Differential clock to external SDRAM.<br>DDR_CK is the true part of the differential signal. DDR_nCk is the complement part. |
| DDR_CKE | O | SDRAM clock enable.<br>0: Disables clock in external SDRAM.<br>1: Enables clock in external SDRAM. |
| DDR_nRAS<br>DDR_nCAS<br>DDR_nWE | O | SDRAM command outputs.<br>DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered. |
| DDR_DM | O | SDRAM data mask outputs.<br>DDR_DM and DDR_UDM are mask signals for data written to the SDRAM. DDR_LDM corresponds to data on DDR_DQ[7:0], and DDR_UDM corresponds to data on DDR_DQ[15:8]. |
| DDR_BA[2:0] | O | SDRAM bank address outputs.<br>DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied. |
| DDR_A[13:0] | O | SDRAM address outputs.<br>Provide row and column addresses to the SDRAM. |
| DDR_DQ[7:0] | I/O | SDRAM data bus. |
| DDR_DQS<br>DDR_DQSn | I/O, Diff | SDRAM differential data strobes.<br>Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. |
| DDR_ODT | O | Control signals for the attached DDR2 SDRAM devices on-die termination. |
| DDR_Rext | ABIAS | External DDR impedance calibration.<br>Connect the pin through an external 240 Ω ±1% resistor to ground. |
| DDR_Vref | ABIAS | Input reference voltage.<br>Provides the input switching reference voltage to the SSTL DDR signals. |

### 10.2.4 General-Purpose Inputs and Outputs

The following table lists the pins associated with general-purpose inputs and outputs. The GPIO pins have an alternate function signal, which is mapped out in the following table. The overlaid functions are

selected by software on a pin-by-pin basis. The parallel interface and MIIM slave interface are enabled depending on the VCORE_CFG settings and override the normal GPIO and alternate functions.

***Table 872 •*** **GPIO Pin Mapping**

| Name | Overlaid Function 1 | Overlaid Function 2 | Parallel Interface | MIIM Slave Interface | Type |
|---|---|---|---|---|---|
| GPIO_0 | SIO_CLK | | | | I/O, PU,ST, 3V |
| GPIO_1 | SIO_LD | | | | I/O, PU,ST, 3V |
| GPIO_2 | SIO_DO | | | | I/O, PU,ST, 3V |
| GPIO_3 | SIO_DI | | | | I/O, PU,ST, 3V |
| GPIO_4 | TACHO | | | | I/O, PU,ST, 3V |
| GPIO_5 | TWI_SCL | PHY0_LED1 | | | I/O, PU,ST, 3V |
| GPIO_6 | TWI_SDA | PHY1_LED1 | | | I/O, PU,ST, 3V |
| GPIO_7 | | PHY2_LED1 | | | I/O, PU,ST, 3V |
| GPIO_8 | EXT_IRQ0 | PHY3_LED1 | | | I/O, PU,ST, 3V |
| GPIO_9 | EXT_IRQ1 | PHY4_LED1 | | | I/O, PU,ST, 3V |
| GPIO_10 | SFP14_SD | PHY5_LED1 | | | I/O, PU,ST, 3V |
| GPIO_11 | SFP15_SD | PHY6_LED1 | | | I/O, PU,ST, 3V |
| GPIO_12 | SFP17_SD | PHY7_LED1 | | | I/O, PU,ST, 3V |
| GPIO_13 | SFP18_SD | PHY8_LED1 | PI_nCS | | I/O, PU,ST, 3V |
| GPIO_14 | SI_nEN1 | PHY9_LED1 | PI_nWR | SLV_ADDR | I/O, PU,ST, 3V |
| GPIO_15 | SI_nEn2 | PHY10_LED1 | PI_nOE | SLV_MDC | I/O, PU,ST, 3V |
| GPIO_16 | SI_nEn3 | PHY11_LED1 | PI_nDone | SLV_MDIO | I/O, PU,ST, 3V |
| GPIO_17 | SFP10_SD | PHY0_LED0 | PI_A0 | | I/O, PU,ST, 3V |
| GPIO_18 | SFP11_SD | PHY2_LED0 | PI_A1 | | I/O, PU,ST, 3V |
| GPIO_19 | SFP12_SD | PHY2_LED0 | PI_A2 | | I/O, PU,ST, 3V |
| GPIO_20 | SFP13_SD | PHY3_LED0 | PI_A3 | | I/O, PU,ST, 3V |
| GPIO_21 | SFP16_SD | PHY4_LED0 | PI_D0 | | I/O, PU,ST, 3V |
| GPIO_22 | SFP19_SD | PHY5_LED0 | PI_D1 | | I/O, PU,ST, 3V |
| GPIO_23 | SFP24_SD | PHY6_LED0 | PI_D2 | | I/O, PU,ST, 3V |
| GPIO_24 | SFP25_SD | PHY7_LED0 | PI_D3 | | I/O, PU,ST, 3V |
| GPIO_25 | SFP20_SD | PHY8_LED0 | PI_D4 | | I/O, PU,ST, 3V |
| GPIO_26 | SFP21_SD | PHY9_LED0 | PI_D5 | | I/O, PU,ST, 3V |
| GPIO_27 | SFP22_SD | PHY10_LED0 | PI_D6 | | I/O, PU,ST, 3V |
| GPIO_28 | SFP23_SD | PHY11_LED0 | PI_D7 | | I/O, PU,ST, 3V |
| GPIO_29 | PWM | | | | I/O, PU,ST, 3V |
| GPIO_30 | UART_TX | | | | I/O, PU,ST, 3V |
| GPIO_31 | UART_RX | | | | I/O, PU,ST, 3V |

## 10.2.5 JTAG Interface

The following table lists the pins associated with the JTAG interface. The JTAG interface can be connected to the boundary scan TAP controller or the internal VCore-III TAP controller for software debug as described for the VCore_ICE_nEn signal.

The JTAG signals are not 5 V tolerant.

***Table 873 •*** **JTAG Interface Pins**

| Name | Type | Description |
|---|---|---|
| JTAG_nTRST | I, PU, ST, 3V | JTAG test reset, active low. For normal device operation, JTAG_nTRST should be pulled low. |
| JTAG_TCK | I, PU, ST, 3V | JTAG clock. |
| JTAG_TDI | I, PU, ST, 3V | JTAG test data in. |
| JTAG_TDO | OZ, 3V | JTAG test data out. |
| JTAG_TMS | I, PU, ST, 3V | JTAG test mode select. |

## 10.2.6 MII Management Interface

The following table lists the pins associated with the MII Management interface.

***Table 874 •*** **MII Management Interface Pins**

| Name | Type | Description |
|---|---|---|
| MDIO | I/O, 3V | Management data input/output.<br>MDIO is a bidirectional signal between a PHY and the device, used to transfer control and status information. Control information is driven by the device synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by the device. |
| MDC | O, 3V | Management data clock.<br>MDC is sourced by the station management entity (the device) to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal. |

## 10.2.7 Miscellaneous Signals

The following table lists the pins associated with a particular interface or facility on the device.

***Table 875 •*** **Miscellaneous Pins**

| Name | Type | Description |
|---|---|---|
| nReset | I, PD, ST, 3V | Global device reset, active low. |
| COMA_MODE | I/O, PU, ST, 3V | When this pin is asserted high, all PHYs are held in a powered-down state.<br>When this pin is deasserted low, all PHYs are powered up and resume normal operation. Additionally, this signal is used to synchronize the operation of multiple devices on the same printed circuit board to provide visual synchronization for LEDs driven from the separate devices. |

*Table 875 •* **Miscellaneous Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| VCORE_CFG[2:0] | I, PD, ST, 3V | Configuration signals for controlling the VCore-III CPU functions. |
| VCore_ICE_nEn | I, PU, 3V | VCore ICE nEn.<br>0: Enables the VCore-III JTAG debug interface over the JTAG interface pins.<br>1: Enables normal IO-JTAG over the JTAG interface. |
| THERMDA | A | Thermal diode anode (p-junction). |
| THERMDC_VSS | A | Thermal diode cathode (n-junction). Connected on-die to $V_{SS}$. |
| EXT_IRQ[1:0](1) | I/O PD, 3V | This pin interrupts inputs or outputs to the internal VCore-III CPU system or to an external processor. Signal polarity is programmable. |
| Reserved_1<br>Reserved_[6:7] | I, PD, ST, 3V | Tie to $V_{DD_IO}$. |
| Reserved_4<br>Reserved_5<br>Reserved_8 | I, PD, ST, 3V | Tie to $V_{SS}$. |
| Reserved_[10:15]<br>Reserved_[17:24]<br>Reserved_[124:127]<br>Reserved_[136:147]<br>Reserved_[201:209]<br>Reserved_[211:221]<br>Reserved_[223]<br>Reserved_[225]<br>Reserved_[232:237]<br>Reserved_[240:248] | I, PD, ST, 3V | Leave floating. |

1. Available as an alternate function on the GPIO_8 and GPIO_9 pins.

## 10.2.8 Parallel Interface

The parallel interface (PI) can operate in a Master mode or a Slave mode according to the VCORE_CFG[1:0] signal settings. In Master mode, the internal VCore-III CPU system controls the PI and can access external peripherals over it. In Slave mode, the PI can be used by an external CPU to access internal device resources.

The PI master and slave mode signals are alternate function signals on GPIO pins. For more information about the GPIO mapping, see Table 872, page 664.

*Table 876 •* **Parallel Interface VCore-III Master Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | OZ, 3V | External address bus.<br>Used for addressing external memory space. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | External data bus. PI_Data0 is LSB. |
| PI_nCS | OZ, 3V | Programmable active low chip selects.<br>PI_nCS is used as default for booting from external memory (typically Flash). |

*Table 876 •* **Parallel Interface VCore-III Master Mode Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| PI_nDone | I, 3V | Acknowledges an operation. Used for external device-paced access operation.<br>Signal polarity is programmable. |
| PI_nOE | OZ, 3V | Active low signal that signals external device to drive data bus during read access. |
| PI_nWR | OZ, 3V | Active low signal that signals external access direction. Read (1) or write (0). |

The following pins are associated with the parallel CPU interface slave mode.

*Table 877 •* **Parallel CPU Interface Slave Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | I, 3V | Internal device register address bus. Controlled by external CPU. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | Data bus.<br>PI_Data[0] is LSB. |
| PI_nCS | I, 3V | Device chip select. |
| PI_nDone | O, 3V | Acknowledges an operation. Signal polarity is programmable. |
| PI_nOE | I, 3V | Signals device to drive data bus during read operations. |
| PI_nWR | I, 3V | Signals access direction. Read (1) or write (0). |

## 10.2.9 Power Supplies and Ground

The following table lists the power supply and ground pins.

*Table 878 •* **Power Supply and Ground Pins**

| Name | Type | Description |
|---|---|---|
| VDD | Power | 1.0 V power supply voltage for core |
| VDD_A | Power | 1.0 V power supply voltage for analog circuits |
| VDD_AL | Power | 1.0 V power supply voltage for analog circuits for twisted pair interface |
| VDD_AH | Power | 2.5 V power supply voltage for analog driver in twisted pair interface |
| VDD_IO | Power | 2.5 V power supply for MII Management interface, parallel CPU interface, and miscellaneous I/Os |
| VDD_IODDR | Power | 1.8 V power supply for DDR interface |
| VDD_VS | Power | 1.0 V or 1.2 V power supply for SerDes and Enhanced SerDes interfaces |
| VSS | Ground | Ground reference |

## 10.2.10 Serial CPU Interface

The serial CPU interface (SI) can be used as a serial slave, master, or boot interface.

As a slave interface, it allows external CPUs to access internal registers. As a master interface, it allows the device to access external devices using a programmable protocol. The serial CPU interface also allows the internal VCore-III CPU system to boot from an attached serial memory device when the VCORE_CFG signals are set appropriately.

The following table lists the pins associated with the serial CPU interface (SI).

***Table 879 •*** **Serial CPU Interface Pins**

| Name | Type | Description |
|---|---|---|
| SI_Clk | I/O, 3V | Slave mode: Input receiving serial interface clock from external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven with clock to external serial memory device. |
| SI_DI | I, 3V | Slave mode: Input receiving serial interface data from external master.<br>Master mode: Input directly read by software from register bit.<br>Boot mode: Input boot data from external serial memory device. |
| SI_DO | OZ, 3V | Slave mode: Output transmitting serial interface data to external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: No function. |
| SI_nEn<br>SI_nEn[3:1](1) | I/O, 3V | Slave mode: Input used to enable SI slave interface.<br>0: Enabled<br>1 = Disabled<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven while booting from EEPROM or serial flash to internal VCore-III CPU system. Released when booting is completed. |

1. Available as an alternate function on the GPIO_16, GPIO_15, and GPIO_14 pins. For more information about GPIO pin mapping, see Table 872, page 664.

## 10.2.11 SerDes Interface

The following pins are associated with the SerDes (SGMII) interface.

***Table 880 •*** **SerDes Interface Pins**

| Name | Type | Description |
|---|---|---|
| SerDes[5:0]_RxP, N | I, Diff, TD | Differential SerDes data inputs. |
| SerDes[5:0]_TxP, N | O, Diff | Differential SerDes data outputs. |

## 10.2.12 Enhanced SerDes Interface

The following pins are associated with the Enhanced SerDes interface.

***Table 881 •*** **Enhanced SerDes Interface Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_E[1:0]_RxP, N | I, Diff, TD | Differential SGMII data inputs. |
| SerDes_E[1:0]_TxP, N | O, Diff | Differential SGMII data outputs. |

## 10.2.13 Twisted Pair Interface

The following pins are associated with the twisted pair interface. The PHYn_LED[1:0] LED control signals associated with the twisted pair interfaces are alternate functions on the GPIOs. For more information about the GPIO pin mapping, see Table 872, page 664.

***Table 882 •*** **Twisted Pair Interface Pins**

| Name | Type | Description |
|---|---|---|
| P0_D0P<br>P1_D0P<br>P2_D0P<br>P3_D0P<br>P4_D0P<br>P5_D0P<br>P6_D0P<br>P7_D0P<br>P8_D0P<br>P9_D0P<br>P10_D0P<br>P11_D0P | $A_{DIFF}$ | Tx/Rx channel A positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1. |
| P0_D0N<br>P1_D0N<br>P2_D0N<br>P3_D0N<br>P4_D0N<br>P5_D0N<br>P6_D0N<br>P7_D0N<br>P8_D0N<br>P9_D0N<br>P10_D0N<br>P11_D0N | $A_{DIFF}$ | Tx/Rx channel A negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2. |
| P0_D1P<br>P1_D1P<br>P2_D1P<br>P3_D1P<br>P4_D1P<br>P5_D1P<br>P6_D1P<br>P7_D1P<br>P8_D1P<br>P9_D1P<br>P10_D1P<br>P11_D1P | $A_{DIFF}$ | Tx/Rx channel B positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3. |
| P0_D1N<br>P1_D1N<br>P2_D1N<br>P3_D1N<br>P4_D1N<br>P5_D1N<br>P6_D1N<br>P7_D1N<br>P8_D1N<br>P9_D1N<br>P10_D1N<br>P11_D1N | $A_{DIFF}$ | Tx/Rx channel B negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6. |

*Table 882 •* **Twisted Pair Interface Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| P0_D2P<br>P1_D2P<br>P2_D2P<br>P3_D2P<br>P4_D2P<br>P5_D2P<br>P6_D2P<br>P7_D2P<br>P8_D2P<br>P9_D2P<br>P10_D2P<br>P11_D2P | $A_{DIFF}$ | Tx/Rx channel C positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes). |
| P0_D2N<br>P1_D2N<br>P2_D2N<br>P3_D2N<br>P4_D2N<br>P5_D2N<br>P6_D2N<br>P7_D2N<br>P8_D2N<br>P9_D2N<br>P10_D2N<br>P11_D2N | $A_{DIFF}$ | Tx/Rx channel C negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes). |
| P0_D3P<br>P1_D3P<br>P2_D3P<br>P3_D3P<br>P4_D3P<br>P5_D3P<br>P6_D3P<br>P7_D3P<br>P8_D3P<br>P9_D3P<br>P10_D3P<br>P11_D3P | $A_{DIFF}$ | Tx/Rx channel D positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes). |
| P0_D3N<br>P1_D3N<br>P2_D3N<br>P3_D3N<br>P4_D3N<br>P5_D3N<br>P6_D3N<br>P7_D3N<br>P8_D3N<br>P9_D3N<br>P10_D3N<br>P11_D3N | $A_{DIFF}$ | Tx/Rx channel D negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes). |

## 10.3 Pins by Number for VSC7425-02

This section provides a numeric list of the VSC7425-02 pins.

| | |
|---|---|
| A2 | P8_D0P |
| A3 | P8_D1P |
| A4 | P8_D2P |
| A5 | P8_D3P |
| A6 | P7_D0P |
| A7 | P7_D1P |
| A8 | P7_D2P |
| A9 | P7_D3P |
| A10 | P6_D0P |
| A11 | P6_D1P |
| A12 | P6_D2P |
| A13 | P6_D3P |
| A14 | P5_D0P |
| A15 | P5_D1P |
| A16 | P5_D2P |
| A17 | P5_D3P |
| A18 | P4_D0P |
| A19 | P4_D1P |
| A20 | P4_D2P |
| A21 | P4_D3P |
| A22 | P3_D0P |
| A23 | P3_D1P |
| A24 | P3_D2P |
| A25 | P3_D3P |
| AA1 | GPIO_7 |
| AA2 | GPIO_6 |
| AA3 | GPIO_5 |
| AA4 | GPIO_4 |
| AA5 | VDD_IO_14 |
| AA6 | Reserved_147 |
| AA7 | Reserved_140 |
| AA8 | RefClk_N |
| AA9 | Reserved_136 |
| AA10 | SerDes5_TxN |
| AA11 | SerDes4_TxN |
| AA12 | VSS_145 |
| AA13 | Reserved_127 |
| AA14 | SerDes3_TxN |
| AA15 | SerDes2_TxN |
| AA16 | VSS_146 |
| AA17 | SerDes_E1_TxN |
| AA18 | SerDes1_TxN |
| AA19 | SerDes0_TxN |
| AA20 | VSS_147 |
| AA21 | SerDes_E0_TxN |
| AA22 | VDD_IODDR_8 |
| AA23 | DDR_nWE |
| AA24 | DDR_BA2 |
| AA25 | DDR_CK |
| AA26 | DDR_CKn |
| AB1 | GPIO_3 |
| AB2 | GPIO_2 |
| AB3 | GPIO_1 |
| AB4 | GPIO_0 |
| AB5 | VDD_IO_15 |
| AB6 | VSS_129 |
| AB7 | VSS_130 |
| AB8 | VSS_131 |
| AB9 | VSS_132 |
| AB10 | VSS_133 |
| AB11 | VSS_134 |
| AB12 | VSS_135 |
| AB13 | VSS_136 |
| AB14 | VSS_137 |
| AB15 | VSS_138 |
| AB16 | VSS_139 |
| AB17 | VSS_140 |
| AB18 | VSS_141 |
| AB19 | VSS_142 |
| AB20 | VSS_143 |
| AB21 | VSS_144 |
| AB22 | VDD_IODDR_9 |
| AB23 | DDR_DQ3 |
| AB24 | DDR_CKE |
| AB25 | DDR_DQ2 |
| AB26 | DDR_DQ5 |
| AC1 | SI_DO |
| AC2 | SI_nEn |
| AC3 | VSS_148 |
| AC4 | VDD_IO_16 |
| AC5 | VDD_IO_17 |
| AC6 | VDD_A_1 |
| AC7 | VDD_A_2 |
| AC8 | VDD_A_3 |
| AC9 | VDD_A_4 |
| AC10 | VDD_A_5 |
| AC11 | VDD_A_6 |
| AC12 | VDD_A_7 |
| AC13 | VDD_A_8 |
| AC14 | VDD_A_9 |
| AC15 | VDD_A_10 |
| AC16 | VDD_A_11 |
| AC17 | VDD_A_12 |
| AC18 | VDD_A_13 |
| AC19 | VDD_A_14 |
| AC20 | VDD_A_15 |
| AC21 | VDD_A_16 |
| AC22 | VDD_IODDR_10 |
| AC23 | DDR_DQ1 |
| AC24 | DDR_DQ4 |
| AC25 | DDR_DQ7 |
| AC26 | DDR_DQ0 |
| AD1 | SI_Clk |
| AD2 | SI_DI |
| AD3 | Reserved_18 |
| AD4 | VDD_IO_18 |
| AD5 | VSS_149 |
| AD6 | VDD_VS_1 |
| AD7 | VDD_VS_2 |
| AD8 | VDD_VS_3 |

Pins by number *(continued)*

| | |
|---|---|
| AD9 | VDD_VS_4 |
| AD10 | VDD_VS_5 |
| AD11 | VDD_VS_6 |
| AD12 | VDD_VS_7 |
| AD13 | VDD_VS_8 |
| AD14 | VDD_VS_9 |
| AD15 | VDD_VS_10 |
| AD16 | VDD_VS_11 |
| AD17 | VDD_VS_12 |
| AD18 | VDD_VS_13 |
| AD19 | VDD_VS_14 |
| AD20 | VDD_VS_15 |
| AD21 | VDD_VS_16 |
| AD22 | VSS_150 |
| AD23 | VDD_IODDR_11 |
| AD24 | DDR_DQ6 |
| AD25 | DDR_DQS |
| AD26 | DDR_DQSn |
| AE1 | VSS_151 |
| AE2 | Reserved_17 |
| AE3 | VDD_IO_19 |
| AE4 | VSS_163 |
| AE5 | VSS_152 |
| AE6 | Reserved_144 |
| AE7 | Reserved_143 |
| AE8 | Reserved_22 |
| AE9 | Reserved_139 |
| AE10 | SerDes5_RxP |
| AE11 | SerDes4_RxP |
| AE12 | VSS_153 |
| AE13 | Reserved_124 |
| AE14 | SerDes3_RxP |
| AE15 | SerDes2_RxP |
| AE16 | VSS_154 |
| AE17 | SerDes_E1_RxP |
| AE18 | SerDes1_RxP |
| AE19 | SerDes0_RxP |
| AE20 | VSS_155 |
| AE21 | SerDes_E0_RxP |
| AE22 | SerDes_Rext_0 |
| AE23 | VSS_156 |
| AE24 | VDD_IODDR_12 |
| AE25 | DDR_DM |
| AE26 | VSS_157 |
| AF2 | VDD_IO_20 |
| AF3 | MDIO |
| AF4 | MDC |
| AF5 | VSS_158 |
| AF6 | Reserved_145 |
| AF7 | Reserved_142 |
| AF8 | Reserved_23 |
| AF9 | Reserved_138 |
| AF10 | SerDes5_RxN |
| AF11 | SerDes4_RxN |
| AF12 | VSS_159 |
| AF13 | Reserved_125 |
| AF14 | SerDes3_RxN |
| AF15 | SerDes2_RxN |
| AF16 | VSS_160 |
| AF17 | SerDes_E1_RxN |
| AF18 | SerDes1_RxN |
| AF19 | SerDes0_RxN |
| AF20 | VSS_161 |
| AF21 | SerDes_E0_RxN |
| AF22 | SerDes_Rext_1 |
| AF23 | VSS_162 |
| AF24 | VDD_IODDR_14 |
| AF25 | VDD_IODDR_13 |
| B1 | VSS_1 |
| B2 | P8_D0N |
| B3 | P8_D1N |
| B4 | P8_D2N |
| B5 | P8_D3N |
| B6 | P7_D0N |
| B7 | P7_D1N |
| B8 | P7_D2N |
| B9 | P7_D3N |
| B10 | P6_D0N |
| B11 | P6_D1N |
| B12 | P6_D2N |
| B13 | P6_D3N |
| B14 | P5_D0N |
| B15 | P5_D1N |
| B16 | P5_D2N |
| B17 | P5_D3N |
| B18 | P4_D0N |
| B19 | P4_D1N |
| B20 | P4_D2N |
| B21 | P4_D3N |
| B22 | P3_D0N |
| B23 | P3_D1N |
| B24 | P3_D2N |
| B25 | P3_D3N |
| B26 | VSS_2 |
| C1 | P9_D3P |
| C2 | P9_D3N |
| C3 | COMA_MODE |
| C4 | nRESET |
| C5 | VDD_IO_21 |
| C6 | Reserved_1 |
| C7 | VCORE_CFG0 |
| C8 | VCORE_CFG1 |
| C9 | VCORE_CFG2 |
| C10 | VCore_ICE_nEn |
| C11 | Reserved_4 |
| C12 | RefClk_Sel0 |
| C13 | RefClk_Sel1 |
| C14 | RefClk_Sel2 |
| C15 | Reserved_8 |
| C16 | Reserved_7 |
| C17 | Reserved_6 |
| C18 | Reserved_5 |
| C19 | Reserved_201 |
| C20 | Reserved_202 |
| C21 | Reserved_203 |
| C22 | THERMDC_VSS |
| C23 | THERMDA |

Pins by number *(continued)*

| | |
|---|---|
| C24 | Reserved_204 |
| C25 | P2_D0N |
| C26 | P2_D0P |
| D1 | P9_D2P |
| D2 | P9_D2N |
| D3 | Reserved_205 |
| D4 | VDD_AH_1 |
| D5 | VDD_AH_2 |
| D6 | Reserved_206 |
| D7 | Reserved_207 |
| D8 | Reserved_208 |
| D9 | Reserved_209 |
| D10 | Reserved_248 |
| D11 | VDD_AH_4 |
| D12 | Reserved_211 |
| D13 | Reserved_13 |
| D14 | Reserved_12 |
| D15 | Reserved_212 |
| D16 | VDD_AH_5 |
| D17 | JTAG_CLK |
| D18 | JTAG_DI |
| D19 | JTAG_DO |
| D20 | JTAG_TMS |
| D21 | JTAG_TRST |
| D22 | Reserved_213 |
| D23 | Reserved_214 |
| D24 | Reserved_215 |
| D25 | P2_D1N |
| D26 | P2_D1P |
| E1 | P9_D1P |
| E2 | P9_D1N |
| E3 | Reserved_216 |
| E4 | VDD_AH_7 |
| E5 | VDD_AH_8 |
| E6 | VDD_IO_1 |
| E7 | VDD_IO_2 |
| E8 | VDD_AH_9 |
| E9 | VDD_AL_1 |
| E10 | VDD_AL_2 |
| E11 | VDD_AH_10 |
| E12 | VDD_AH_11 |
| E13 | Ref_rext_1 |
| E14 | Ref_filt_1 |
| E15 | VDD_AH_12 |
| E16 | VDD_AH_13 |
| E17 | VDD_AL_3 |
| E18 | VDD_AL_4 |
| E19 | VDD_AH_14 |
| E20 | VDD_IO_3 |
| E21 | VDD_IO_4 |
| E22 | VDD_AH_15 |
| E23 | VDD_AH_16 |
| E24 | Reserved_217 |
| E25 | P2_D2N |
| E26 | P2_D2P |
| F1 | P9_D0P |
| F2 | P9_D0N |
| F3 | Reserved_218 |
| F4 | VDD_AH_17 |
| F5 | VDD_AH_18 |
| F6 | VDD_IO_5 |
| F7 | VDD_AH_3 |
| F8 | VDD_AH_19 |
| F9 | VDD_AL_5 |
| F10 | VDD_AL_6 |
| F11 | VDD_AH_20 |
| F12 | VDD_AH_21 |
| F13 | Reserved_219 |
| F14 | Reserved_220 |
| F15 | VDD_AH_22 |
| F16 | VDD_AH_23 |
| F17 | VDD_AL_7 |
| F18 | VDD_AL_8 |
| F19 | VDD_AH_24 |
| F20 | VDD_AH_6 |
| F21 | VDD_IO_6 |
| F22 | VDD_AH_25 |
| F23 | VDD_AH_26 |
| F24 | Reserved_221 |
| F25 | P2_D3N |
| F26 | P2_D3P |
| G1 | P10_D3P |
| G2 | P10_D3N |
| G3 | VSS_3 |
| G4 | Reserved_15 |
| G5 | VSS_4 |
| G6 | VDD_1 |
| G7 | VDD_2 |
| G8 | VDD_3 |
| G9 | VDD_AL_9 |
| G10 | VDD_AL_10 |
| G11 | VDD_4 |
| G12 | VDD_5 |
| G13 | Reserved_247 |
| G14 | Reserved_223 |
| G15 | VDD_6 |
| G16 | VDD_7 |
| G17 | VDD_AL_11 |
| G18 | VDD_AL_12 |
| G19 | VDD_8 |
| G20 | VDD_9 |
| G21 | VDD_10 |
| G22 | VSS_5 |
| G23 | Reserved_10 |
| G24 | VSS_6 |
| G25 | P1_D0N |
| G26 | P1_D0P |
| H1 | P10_D2P |
| H2 | P10_D2N |
| H3 | VSS_7 |
| H4 | Reserved_14 |
| H5 | VSS_8 |
| H6 | VDD_11 |
| H7 | VDD_12 |
| H8 | VDD_13 |
| H9 | VDD_14 |
| H10 | VDD_15 |

Pins by number *(continued)*

| | |
|---|---|
| H11 | VDD_16 |
| H12 | VDD_17 |
| H13 | Reserved_246 |
| H14 | Reserved_225 |
| H15 | VDD_18 |
| H16 | VDD_19 |
| H17 | VDD_20 |
| H18 | VDD_21 |
| H19 | VDD_22 |
| H20 | VDD_23 |
| H21 | VDD_24 |
| H22 | VSS_9 |
| H23 | Reserved_11 |
| H24 | VSS_10 |
| H25 | P1_D1N |
| H26 | P1_D1P |
| J1 | P10_D1P |
| J2 | P10_D1N |
| J3 | VDD_AH_27 |
| J4 | VDD_AH_28 |
| J5 | VDD_AL_13 |
| J6 | VDD_AL_14 |
| J7 | VDD_AL_15 |
| J8 | Reserved_240 |
| J9 | Reserved_241 |
| J10 | Reserved_242 |
| J11 | Reserved_243 |
| J12 | Reserved_244 |
| J13 | Reserved_245 |
| J14 | Reserved_232 |
| J15 | Reserved_233 |
| J16 | Reserved_234 |
| J17 | Reserved_235 |
| J18 | Reserved_236 |
| J19 | Reserved_237 |
| J20 | VDD_AL_16 |
| J21 | VDD_AL_17 |
| J22 | VDD_AL_18 |
| J23 | VDD_AH_29 |
| J24 | VDD_AH_30 |
| J25 | P1_D2N |
| J26 | P1_D2P |
| K1 | P10_D0P |
| K2 | P10_D0N |
| K3 | VSS_11 |
| K4 | Ref_rext_2 |
| K5 | VDD_AL_19 |
| K6 | VDD_AL_20 |
| K7 | VDD_AL_21 |
| K8 | VSS_12 |
| K9 | VSS_13 |
| K10 | VSS_14 |
| K11 | VSS_15 |
| K12 | VSS_16 |
| K13 | VSS_17 |
| K14 | VSS_18 |
| K15 | VSS_19 |
| K16 | VSS_20 |
| K17 | VSS_21 |
| K18 | VSS_22 |
| K19 | VSS_23 |
| K20 | VDD_AL_22 |
| K21 | VDD_AL_23 |
| K22 | VDD_AL_24 |
| K23 | Ref_rext_0 |
| K24 | VSS_24 |
| K25 | P1_D3N |
| K26 | P1_D3P |
| L1 | P11_D3P |
| L2 | P11_D3N |
| L3 | VSS_25 |
| L4 | Ref_filt_2 |
| L5 | VSS_26 |
| L6 | VDD_25 |
| L7 | VDD_26 |
| L8 | VSS_27 |
| L9 | VSS_28 |
| L10 | VSS_29 |
| L11 | VSS_30 |
| L12 | VSS_31 |
| L13 | VSS_32 |
| L14 | VSS_33 |
| L15 | VSS_34 |
| L16 | VSS_35 |
| L17 | VSS_36 |
| L18 | VSS_37 |
| L19 | VSS_38 |
| L20 | VDD_27 |
| L21 | VDD_28 |
| L22 | VSS_39 |
| L23 | Ref_filt_0 |
| L24 | VSS_40 |
| L25 | P0_D0N |
| L26 | P0_D0P |
| M1 | P11_D2P |
| M2 | P11_D2N |
| M3 | VDD_AH_31 |
| M4 | VDD_AH_32 |
| M5 | VDD_AH_33 |
| M6 | VDD_29 |
| M7 | VDD_30 |
| M8 | VSS_41 |
| M9 | VSS_42 |
| M10 | VSS_43 |
| M11 | VSS_44 |
| M12 | VSS_45 |
| M13 | VSS_46 |
| M14 | VSS_47 |
| M15 | VSS_48 |
| M16 | VSS_49 |
| M17 | VSS_50 |
| M18 | VSS_51 |
| M19 | VSS_52 |
| M20 | VDD_31 |
| M21 | VDD_32 |
| M22 | VDD_AH_34 |
| M23 | VDD_AH_35 |

Pins by number *(continued)*

| | |
|---|---|
| M24 | VDD_AH_36 |
| M25 | P0_D1N |
| M26 | P0_D1P |
| N1 | P11_D1P |
| N2 | P11_D1N |
| N3 | VSS_53 |
| N4 | VSS_54 |
| N5 | VSS_55 |
| N6 | VDD_33 |
| N7 | VDD_34 |
| N8 | VSS_56 |
| N9 | VSS_57 |
| N10 | VSS_58 |
| N11 | VSS_59 |
| N12 | VSS_60 |
| N13 | VSS_61 |
| N14 | VSS_62 |
| N15 | VSS_63 |
| N16 | VSS_64 |
| N17 | VSS_65 |
| N18 | VSS_66 |
| N19 | VSS_67 |
| N20 | VDD_35 |
| N21 | VDD_36 |
| N22 | VSS_68 |
| N23 | VSS_69 |
| N24 | VSS_70 |
| N25 | P0_D2N |
| N26 | P0_D2P |
| P1 | P11_D0P |
| P2 | P11_D0N |
| P3 | VSS_71 |
| P4 | Reserved_24 |
| P5 | VDD_IO_7 |
| P6 | VDD_37 |
| P7 | VDD_38 |
| P8 | VSS_72 |
| P9 | VSS_73 |
| P10 | VSS_74 |

| | |
|---|---|
| P11 | VSS_75 |
| P12 | VSS_76 |
| P13 | VSS_77 |
| P14 | VSS_78 |
| P15 | VSS_79 |
| P16 | VSS_80 |
| P17 | VSS_81 |
| P18 | VSS_82 |
| P19 | VSS_83 |
| P20 | VDD_39 |
| P21 | VDD_40 |
| P22 | VDD_IODDR_1 |
| P23 | VSS_84 |
| P24 | VSS_85 |
| P25 | P0_D3N |
| P26 | P0_D3P |
| R1 | GPIO_31 |
| R2 | GPIO_30 |
| R3 | GPIO_29 |
| R4 | GPIO_28 |
| R5 | VDD_IO_8 |
| R6 | VDD_41 |
| R7 | VDD_42 |
| R8 | VSS_86 |
| R9 | VSS_87 |
| R10 | VSS_88 |
| R11 | VSS_89 |
| R12 | VSS_90 |
| R13 | VSS_91 |
| R14 | VSS_92 |
| R15 | VSS_93 |
| R16 | VSS_94 |
| R17 | VSS_95 |
| R18 | VSS_96 |
| R19 | VSS_97 |
| R20 | VDD_43 |
| R21 | VDD_44 |
| R22 | VDD_IODDR_2 |
| R23 | Reserved_20 |

| | |
|---|---|
| R24 | Reserved_19 |
| R25 | DDR_Rext |
| R26 | DDR_Vref |
| T1 | GPIO_27 |
| T2 | GPIO_26 |
| T3 | GPIO_25 |
| T4 | GPIO_24 |
| T5 | VDD_IO_9 |
| T6 | VDD_45 |
| T7 | VDD_46 |
| T8 | VSS_98 |
| T9 | VSS_99 |
| T10 | VSS_100 |
| T11 | VSS_101 |
| T12 | VSS_102 |
| T13 | VSS_103 |
| T14 | VSS_104 |
| T15 | VSS_105 |
| T16 | VSS_106 |
| T17 | VSS_107 |
| T18 | VSS_108 |
| T19 | VSS_109 |
| T20 | VDD_47 |
| T21 | VDD_48 |
| T22 | VDD_IODDR_3 |
| T23 | Reserved_21 |
| T24 | DDR_A13 |
| T25 | DDR_A12 |
| T26 | DDR_A11 |
| U1 | GPIO_23 |
| U2 | GPIO_22 |
| U3 | GPIO_21 |
| U4 | GPIO_20 |
| U5 | VDD_IO_10 |
| U6 | VSS_110 |
| U7 | VSS_111 |
| U8 | VSS_112 |
| U9 | VSS_113 |
| U10 | VSS_114 |

Pins by number *(continued)*

| | |
|---|---|
| U11 | VSS_115 |
| U12 | VSS_116 |
| U13 | VSS_117 |
| U14 | VSS_118 |
| U15 | VSS_119 |
| U16 | VSS_120 |
| U17 | VSS_121 |
| U18 | VSS_122 |
| U19 | VSS_123 |
| U20 | VSS_124 |
| U21 | VSS_125 |
| U22 | VDD_IODDR_4 |
| U23 | DDR_A7 |
| U24 | DDR_A9 |
| U25 | DDR_A6 |
| U26 | DDR_A8 |
| V1 | GPIO_19 |
| V2 | GPIO_18 |
| V3 | GPIO_17 |
| V4 | GPIO_16 |
| V5 | VDD_IO_11 |
| V6 | VDD_49 |
| V7 | VDD_50 |
| V8 | VDD_51 |
| V9 | VDD_52 |
| V10 | VDD_53 |
| V11 | VDD_54 |
| V12 | VDD_55 |
| V13 | VDD_56 |
| V14 | VDD_57 |
| V15 | VDD_58 |
| V16 | VDD_59 |
| V17 | VDD_60 |
| V18 | VDD_61 |
| V19 | VDD_62 |
| V20 | VDD_63 |
| V21 | VDD_64 |
| V22 | VDD_IODDR_5 |
| V23 | DDR_A3 |
| V24 | DDR_A5 |
| V25 | DDR_A2 |
| V26 | DDR_A4 |
| W1 | GPIO_15 |
| W2 | GPIO_14 |
| W3 | GPIO_13 |
| W4 | GPIO_12 |
| W5 | VDD_IO_12 |
| W6 | VDD_65 |
| W7 | VDD_66 |
| W8 | VDD_67 |
| W9 | VDD_68 |
| W10 | VDD_69 |
| W11 | VDD_70 |
| W12 | VDD_71 |
| W13 | VDD_72 |
| W14 | VDD_73 |
| W15 | VDD_74 |
| W16 | VDD_75 |
| W17 | VDD_76 |
| W18 | VDD_77 |
| W19 | VDD_78 |
| W20 | VDD_79 |
| W21 | VDD_80 |
| W22 | VDD_IODDR_6 |
| W23 | DDR_A10 |
| W24 | DDR_A1 |
| W25 | DDR_nCAS |
| W26 | DDR_A0 |
| Y1 | GPIO_11 |
| Y2 | GPIO_10 |
| Y3 | GPIO_9 |
| Y4 | GPIO_8 |
| Y5 | VDD_IO_13 |
| Y6 | Reserved_146 |
| Y7 | Reserved_141 |
| Y8 | RefClk_P |
| Y9 | Reserved_137 |
| Y10 | SerDes5_TxP |
| Y11 | SerDes4_TxP |
| Y12 | VSS_126 |
| Y13 | Reserved_126 |
| Y14 | SerDes3_TxP |
| Y15 | SerDes2_TxP |
| Y16 | VSS_127 |
| Y17 | SerDes_E1_TxP |
| Y18 | SerDes1_TxP |
| Y19 | SerDes0_TxP |
| Y20 | VSS_128 |
| Y21 | SerDes_E0_TxP |
| Y22 | VDD_IODDR_7 |
| Y23 | DDR_BA0 |
| Y24 | DDR_BA1 |
| Y25 | DDR_ODT |
| Y26 | DDR_nRAS |

## 10.4 Pins by Name for VSC7425-02

This section provides an alphabetical list of the VSC7425-02 pins.

| | |
|---|---|
| COMA_MODE | C3 |
| DDR_A0 | W26 |
| DDR_A1 | W24 |
| DDR_A2 | V25 |
| DDR_A3 | V23 |
| DDR_A4 | V26 |
| DDR_A5 | V24 |
| DDR_A6 | U25 |
| DDR_A7 | U23 |
| DDR_A8 | U26 |
| DDR_A9 | U24 |
| DDR_A10 | W23 |
| DDR_A11 | T26 |
| DDR_A12 | T25 |
| DDR_A13 | T24 |
| DDR_BA0 | Y23 |
| DDR_BA1 | Y24 |
| DDR_BA2 | AA24 |
| DDR_CK | AA25 |
| DDR_CKE | AB24 |
| DDR_CKn | AA26 |
| DDR_DM | AE25 |
| DDR_DQ0 | AC26 |
| DDR_DQ1 | AC23 |
| DDR_DQ2 | AB25 |
| DDR_DQ3 | AB23 |
| DDR_DQ4 | AC24 |
| DDR_DQ5 | AB26 |
| DDR_DQ6 | AD24 |
| DDR_DQ7 | AC25 |
| DDR_DQS | AD25 |
| DDR_DQSn | AD26 |
| DDR_nCAS | W25 |
| DDR_nRAS | Y26 |
| DDR_nWE | AA23 |
| DDR_ODT | Y25 |
| DDR_Rext | R25 |
| DDR_Vref | R26 |
| GPIO_0 | AB4 |
| GPIO_1 | AB3 |
| GPIO_2 | AB2 |
| GPIO_3 | AB1 |
| GPIO_4 | AA4 |
| GPIO_5 | AA3 |
| GPIO_6 | AA2 |
| GPIO_7 | AA1 |
| GPIO_8 | Y4 |
| GPIO_9 | Y3 |
| GPIO_10 | Y2 |
| GPIO_11 | Y1 |
| GPIO_12 | W4 |
| GPIO_13 | W3 |
| GPIO_14 | W2 |
| GPIO_15 | W1 |
| GPIO_16 | V4 |
| GPIO_17 | V3 |
| GPIO_18 | V2 |
| GPIO_19 | V1 |
| GPIO_20 | U4 |
| GPIO_21 | U3 |
| GPIO_22 | U2 |
| GPIO_23 | U1 |
| GPIO_24 | T4 |
| GPIO_25 | T3 |
| GPIO_26 | T2 |
| GPIO_27 | T1 |
| GPIO_28 | R4 |
| GPIO_29 | R3 |
| GPIO_30 | R2 |
| GPIO_31 | R1 |
| JTAG_CLK | D17 |
| JTAG_DI | D18 |
| JTAG_DO | D19 |
| JTAG_TMS | D20 |
| JTAG_TRST | D21 |
| MDC | AF4 |
| MDIO | AF3 |
| nRESET | C4 |
| P0_D0N | L25 |
| P0_D0P | L26 |
| P0_D1N | M25 |
| P0_D1P | M26 |
| P0_D2N | N25 |
| P0_D2P | N26 |
| P0_D3N | P25 |
| P0_D3P | P26 |
| P1_D0N | G25 |
| P1_D0P | G26 |
| P1_D1N | H25 |
| P1_D1P | H26 |
| P1_D2N | J25 |
| P1_D2P | J26 |
| P1_D3N | K25 |
| P1_D3P | K26 |
| P2_D0N | C25 |
| P2_D0P | C26 |
| P2_D1N | D25 |
| P2_D1P | D26 |
| P2_D2N | E25 |
| P2_D2P | E26 |
| P2_D3N | F25 |
| P2_D3P | F26 |
| P3_D0N | B22 |
| P3_D0P | A22 |
| P3_D1N | B23 |
| P3_D1P | A23 |
| P3_D2N | B24 |
| P3_D2P | A24 |
| P3_D3N | B25 |
| P3_D3P | A25 |

Pins by name *(continued)*

| | |
|---|---|
| P4_D0N | B18 |
| P4_D0P | A18 |
| P4_D1N | B19 |
| P4_D1P | A19 |
| P4_D2N | B20 |
| P4_D2P | A20 |
| P4_D3N | B21 |
| P4_D3P | A21 |
| P5_D0N | B14 |
| P5_D0P | A14 |
| P5_D1N | B15 |
| P5_D1P | A15 |
| P5_D2N | B16 |
| P5_D2P | A16 |
| P5_D3N | B17 |
| P5_D3P | A17 |
| P6_D0N | B10 |
| P6_D0P | A10 |
| P6_D1N | B11 |
| P6_D1P | A11 |
| P6_D2N | B12 |
| P6_D2P | A12 |
| P6_D3N | B13 |
| P6_D3P | A13 |
| P7_D0N | B6 |
| P7_D0P | A6 |
| P7_D1N | B7 |
| P7_D1P | A7 |
| P7_D2N | B8 |
| P7_D2P | A8 |
| P7_D3N | B9 |
| P7_D3P | A9 |
| P8_D0N | B2 |
| P8_D0P | A2 |
| P8_D1N | B3 |
| P8_D1P | A3 |
| P8_D2N | B4 |
| P8_D2P | A4 |
| P8_D3N | B5 |
| P8_D3P | A5 |
| P9_D0N | F2 |
| P9_D0P | F1 |
| P9_D1N | E2 |
| P9_D1P | E1 |
| P9_D2N | D2 |
| P9_D2P | D1 |
| P9_D3N | C2 |
| P9_D3P | C1 |
| P10_D0N | K2 |
| P10_D0P | K1 |
| P10_D1N | J2 |
| P10_D1P | J1 |
| P10_D2N | H2 |
| P10_D2P | H1 |
| P10_D3N | G2 |
| P10_D3P | G1 |
| P11_D0N | P2 |
| P11_D0P | P1 |
| P11_D1N | N2 |
| P11_D1P | N1 |
| P11_D2N | M2 |
| P11_D2P | M1 |
| P11_D3N | L2 |
| P11_D3P | L1 |
| Ref_filt_0 | L23 |
| Ref_filt_1 | E14 |
| Ref_filt_2 | L4 |
| Ref_rext_0 | K23 |
| Ref_rext_1 | E13 |
| Ref_rext_2 | K4 |
| RefClk_N | AA8 |
| RefClk_P | Y8 |
| RefClk_Sel0 | C12 |
| RefClk_Sel1 | C13 |
| RefClk_Sel2 | C14 |
| Reserved_1 | C6 |
| Reserved_4 | C11 |
| Reserved_5 | C18 |
| Reserved_6 | C17 |
| Reserved_7 | C16 |
| Reserved_8 | C15 |
| Reserved_10 | G23 |
| Reserved_11 | H23 |
| Reserved_12 | D14 |
| Reserved_13 | D13 |
| Reserved_14 | H4 |
| Reserved_15 | G4 |
| Reserved_17 | AE2 |
| Reserved_18 | AD3 |
| Reserved_19 | R24 |
| Reserved_20 | R23 |
| Reserved_21 | T23 |
| Reserved_22 | AE8 |
| Reserved_23 | AF8 |
| Reserved_24 | P4 |
| Reserved_124 | AE13 |
| Reserved_125 | AF13 |
| Reserved_126 | Y13 |
| Reserved_127 | AA13 |
| Reserved_136 | AA9 |
| Reserved_137 | Y9 |
| Reserved_138 | AF9 |
| Reserved_139 | AE9 |
| Reserved_140 | AA7 |
| Reserved_141 | Y7 |
| Reserved_142 | AF7 |
| Reserved_143 | AE7 |
| Reserved_144 | AE6 |
| Reserved_145 | AF6 |
| Reserved_146 | Y6 |
| Reserved_147 | AA6 |
| Reserved_201 | C19 |
| Reserved_202 | C20 |
| Reserved_203 | C21 |
| Reserved_204 | C24 |
| Reserved_205 | D3 |
| Reserved_206 | D6 |

Pins by name *(continued)*

| | |
|---|---|
| Reserved_207 | D7 |
| Reserved_208 | D8 |
| Reserved_209 | D9 |
| Reserved_211 | D12 |
| Reserved_212 | D15 |
| Reserved_213 | D22 |
| Reserved_214 | D23 |
| Reserved_215 | D24 |
| Reserved_216 | E3 |
| Reserved_217 | E24 |
| Reserved_218 | F3 |
| Reserved_219 | F13 |
| Reserved_220 | F14 |
| Reserved_221 | F24 |
| Reserved_223 | G14 |
| Reserved_225 | H14 |
| Reserved_232 | J14 |
| Reserved_233 | J15 |
| Reserved_234 | J16 |
| Reserved_235 | J17 |
| Reserved_236 | J18 |
| Reserved_237 | J19 |
| Reserved_240 | J8 |
| Reserved_241 | J9 |
| Reserved_242 | J10 |
| Reserved_243 | J11 |
| Reserved_244 | J12 |
| Reserved_245 | J13 |
| Reserved_246 | H13 |
| Reserved_247 | G13 |
| Reserved_248 | D10 |
| SerDes_E0_RxN | AF21 |
| SerDes_E0_RxP | AE21 |
| SerDes_E0_TxN | AA21 |
| SerDes_E0_TxP | Y21 |
| SerDes_E1_RxN | AF17 |
| SerDes_E1_RxP | AE17 |
| SerDes_E1_TxN | AA17 |
| SerDes_E1_TxP | Y17 |
| SerDes_Rext_0 | AE22 |
| SerDes_Rext_1 | AF22 |
| SerDes0_RxN | AF19 |
| SerDes0_RxP | AE19 |
| SerDes0_TxN | AA19 |
| SerDes0_TxP | Y19 |
| SerDes1_RxN | AF18 |
| SerDes1_RxP | AE18 |
| SerDes1_TxN | AA18 |
| SerDes1_TxP | Y18 |
| SerDes2_RxN | AF15 |
| SerDes2_RxP | AE15 |
| SerDes2_TxN | AA15 |
| SerDes2_TxP | Y15 |
| SerDes3_RxN | AF14 |
| SerDes3_RxP | AE14 |
| SerDes3_TxN | AA14 |
| SerDes3_TxP | Y14 |
| SerDes4_RxN | AF11 |
| SerDes4_RxP | AE11 |
| SerDes4_TxN | AA11 |
| SerDes4_TxP | Y11 |
| SerDes5_RxN | AF10 |
| SerDes5_RxP | AE10 |
| SerDes5_TxN | AA10 |
| SerDes5_TxP | Y10 |
| SI_Clk | AD1 |
| SI_DI | AD2 |
| SI_DO | AC1 |
| SI_nEn | AC2 |
| THERMDA | C23 |
| THERMDC_VSS | C22 |
| VCORE_CFG0 | C7 |
| VCORE_CFG1 | C8 |
| VCORE_CFG2 | C9 |
| VCore_ICE_nEn | C10 |
| VDD_1 | G6 |
| VDD_2 | G7 |
| VDD_3 | G8 |
| VDD_4 | G11 |
| VDD_5 | G12 |
| VDD_6 | G15 |
| VDD_7 | G16 |
| VDD_8 | G19 |
| VDD_9 | G20 |
| VDD_10 | G21 |
| VDD_11 | H6 |
| VDD_12 | H7 |
| VDD_13 | H8 |
| VDD_14 | H9 |
| VDD_15 | H10 |
| VDD_16 | H11 |
| VDD_17 | H12 |
| VDD_18 | H15 |
| VDD_19 | H16 |
| VDD_20 | H17 |
| VDD_21 | H18 |
| VDD_22 | H19 |
| VDD_23 | H20 |
| VDD_24 | H21 |
| VDD_25 | L6 |
| VDD_26 | L7 |
| VDD_27 | L20 |
| VDD_28 | L21 |
| VDD_29 | M6 |
| VDD_30 | M7 |
| VDD_31 | M20 |
| VDD_32 | M21 |
| VDD_33 | N6 |
| VDD_34 | N7 |
| VDD_35 | N20 |
| VDD_36 | N21 |
| VDD_37 | P6 |
| VDD_38 | P7 |
| VDD_39 | P20 |
| VDD_40 | P21 |
| VDD_41 | R6 |
| VDD_42 | R7 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_43 | R20 |
| VDD_44 | R21 |
| VDD_45 | T6 |
| VDD_46 | T7 |
| VDD_47 | T20 |
| VDD_48 | T21 |
| VDD_49 | V6 |
| VDD_50 | V7 |
| VDD_51 | V8 |
| VDD_52 | V9 |
| VDD_53 | V10 |
| VDD_54 | V11 |
| VDD_55 | V12 |
| VDD_56 | V13 |
| VDD_57 | V14 |
| VDD_58 | V15 |
| VDD_59 | V16 |
| VDD_60 | V17 |
| VDD_61 | V18 |
| VDD_62 | V19 |
| VDD_63 | V20 |
| VDD_64 | V21 |
| VDD_65 | W6 |
| VDD_66 | W7 |
| VDD_67 | W8 |
| VDD_68 | W9 |
| VDD_69 | W10 |
| VDD_70 | W11 |
| VDD_71 | W12 |
| VDD_72 | W13 |
| VDD_73 | W14 |
| VDD_74 | W15 |
| VDD_75 | W16 |
| VDD_76 | W17 |
| VDD_77 | W18 |
| VDD_78 | W19 |
| VDD_79 | W20 |
| VDD_80 | W21 |
| VDD_A_1 | AC6 |

| | |
|---|---|
| VDD_A_2 | AC7 |
| VDD_A_3 | AC8 |
| VDD_A_4 | AC9 |
| VDD_A_5 | AC10 |
| VDD_A_6 | AC11 |
| VDD_A_7 | AC12 |
| VDD_A_8 | AC13 |
| VDD_A_9 | AC14 |
| VDD_A_10 | AC15 |
| VDD_A_11 | AC16 |
| VDD_A_12 | AC17 |
| VDD_A_13 | AC18 |
| VDD_A_14 | AC19 |
| VDD_A_15 | AC20 |
| VDD_A_16 | AC21 |
| VDD_AH_1 | D4 |
| VDD_AH_2 | D5 |
| VDD_AH_3 | F7 |
| VDD_AH_4 | D11 |
| VDD_AH_5 | D16 |
| VDD_AH_6 | F20 |
| VDD_AH_7 | E4 |
| VDD_AH_8 | E5 |
| VDD_AH_9 | E8 |
| VDD_AH_10 | E11 |
| VDD_AH_11 | E12 |
| VDD_AH_12 | E15 |
| VDD_AH_13 | E16 |
| VDD_AH_14 | E19 |
| VDD_AH_15 | E22 |
| VDD_AH_16 | E23 |
| VDD_AH_17 | F4 |
| VDD_AH_18 | F5 |
| VDD_AH_19 | F8 |
| VDD_AH_20 | F11 |
| VDD_AH_21 | F12 |
| VDD_AH_22 | F15 |
| VDD_AH_23 | F16 |
| VDD_AH_24 | F19 |

| | |
|---|---|
| VDD_AH_25 | F22 |
| VDD_AH_26 | F23 |
| VDD_AH_27 | J3 |
| VDD_AH_28 | J4 |
| VDD_AH_29 | J23 |
| VDD_AH_30 | J24 |
| VDD_AH_31 | M3 |
| VDD_AH_32 | M4 |
| VDD_AH_33 | M5 |
| VDD_AH_34 | M22 |
| VDD_AH_35 | M23 |
| VDD_AH_36 | M24 |
| VDD_AL_1 | E9 |
| VDD_AL_2 | E10 |
| VDD_AL_3 | E17 |
| VDD_AL_4 | E18 |
| VDD_AL_5 | F9 |
| VDD_AL_6 | F10 |
| VDD_AL_7 | F17 |
| VDD_AL_8 | F18 |
| VDD_AL_9 | G9 |
| VDD_AL_10 | G10 |
| VDD_AL_11 | G17 |
| VDD_AL_12 | G18 |
| VDD_AL_13 | J5 |
| VDD_AL_14 | J6 |
| VDD_AL_15 | J7 |
| VDD_AL_16 | J20 |
| VDD_AL_17 | J21 |
| VDD_AL_18 | J22 |
| VDD_AL_19 | K5 |
| VDD_AL_20 | K6 |
| VDD_AL_21 | K7 |
| VDD_AL_22 | K20 |
| VDD_AL_23 | K21 |
| VDD_AL_24 | K22 |
| VDD_IO_1 | E6 |
| VDD_IO_2 | E7 |
| VDD_IO_3 | E20 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_IO_4 | E21 |
| VDD_IO_5 | F6 |
| VDD_IO_6 | F21 |
| VDD_IO_7 | P5 |
| VDD_IO_8 | R5 |
| VDD_IO_9 | T5 |
| VDD_IO_10 | U5 |
| VDD_IO_11 | V5 |
| VDD_IO_12 | W5 |
| VDD_IO_13 | Y5 |
| VDD_IO_14 | AA5 |
| VDD_IO_15 | AB5 |
| VDD_IO_16 | AC4 |
| VDD_IO_17 | AC5 |
| VDD_IO_18 | AD4 |
| VDD_IO_19 | AE3 |
| VDD_IO_20 | AF2 |
| VDD_IO_21 | C5 |
| VDD_IODDR_1 | P22 |
| VDD_IODDR_2 | R22 |
| VDD_IODDR_3 | T22 |
| VDD_IODDR_4 | U22 |
| VDD_IODDR_5 | V22 |
| VDD_IODDR_6 | W22 |
| VDD_IODDR_7 | Y22 |
| VDD_IODDR_8 | AA22 |
| VDD_IODDR_9 | AB22 |
| VDD_IODDR_10 | AC22 |
| VDD_IODDR_11 | AD23 |
| VDD_IODDR_12 | AE24 |
| VDD_IODDR_13 | AF25 |
| VDD_IODDR_14 | AF24 |
| VDD_VS_1 | AD6 |
| VDD_VS_2 | AD7 |
| VDD_VS_3 | AD8 |
| VDD_VS_4 | AD9 |
| VDD_VS_5 | AD10 |
| VDD_VS_6 | AD11 |
| VDD_VS_7 | AD12 |
| VDD_VS_8 | AD13 |
| VDD_VS_9 | AD14 |
| VDD_VS_10 | AD15 |
| VDD_VS_11 | AD16 |
| VDD_VS_12 | AD17 |
| VDD_VS_13 | AD18 |
| VDD_VS_14 | AD19 |
| VDD_VS_15 | AD20 |
| VDD_VS_16 | AD21 |
| VSS_1 | B1 |
| VSS_2 | B26 |
| VSS_3 | G3 |
| VSS_4 | G5 |
| VSS_5 | G22 |
| VSS_6 | G24 |
| VSS_7 | H3 |
| VSS_8 | H5 |
| VSS_9 | H22 |
| VSS_10 | H24 |
| VSS_11 | K3 |
| VSS_12 | K8 |
| VSS_13 | K9 |
| VSS_14 | K10 |
| VSS_15 | K11 |
| VSS_16 | K12 |
| VSS_17 | K13 |
| VSS_18 | K14 |
| VSS_19 | K15 |
| VSS_20 | K16 |
| VSS_21 | K17 |
| VSS_22 | K18 |
| VSS_23 | K19 |
| VSS_24 | K24 |
| VSS_25 | L3 |
| VSS_26 | L5 |
| VSS_27 | L8 |
| VSS_28 | L9 |
| VSS_29 | L10 |
| VSS_30 | L11 |
| VSS_31 | L12 |
| VSS_32 | L13 |
| VSS_33 | L14 |
| VSS_34 | L15 |
| VSS_35 | L16 |
| VSS_36 | L17 |
| VSS_37 | L18 |
| VSS_38 | L19 |
| VSS_39 | L22 |
| VSS_40 | L24 |
| VSS_41 | M8 |
| VSS_42 | M9 |
| VSS_43 | M10 |
| VSS_44 | M11 |
| VSS_45 | M12 |
| VSS_46 | M13 |
| VSS_47 | M14 |
| VSS_48 | M15 |
| VSS_49 | M16 |
| VSS_50 | M17 |
| VSS_51 | M18 |
| VSS_52 | M19 |
| VSS_53 | N3 |
| VSS_54 | N4 |
| VSS_55 | N5 |
| VSS_56 | N8 |
| VSS_57 | N9 |
| VSS_58 | N10 |
| VSS_59 | N11 |
| VSS_60 | N12 |
| VSS_61 | N13 |
| VSS_62 | N14 |
| VSS_63 | N15 |
| VSS_64 | N16 |
| VSS_65 | N17 |
| VSS_66 | N18 |
| VSS_67 | N19 |
| VSS_68 | N22 |
| VSS_69 | N23 |

Pins by name *(continued)*

| | |
|---|---|
| VSS_70 | N24 |
| VSS_71 | P3 |
| VSS_72 | P8 |
| VSS_73 | P9 |
| VSS_74 | P10 |
| VSS_75 | P11 |
| VSS_76 | P12 |
| VSS_77 | P13 |
| VSS_78 | P14 |
| VSS_79 | P15 |
| VSS_80 | P16 |
| VSS_81 | P17 |
| VSS_82 | P18 |
| VSS_83 | P19 |
| VSS_84 | P23 |
| VSS_85 | P24 |
| VSS_86 | R8 |
| VSS_87 | R9 |
| VSS_88 | R10 |
| VSS_89 | R11 |
| VSS_90 | R12 |
| VSS_91 | R13 |
| VSS_92 | R14 |
| VSS_93 | R15 |
| VSS_94 | R16 |
| VSS_95 | R17 |
| VSS_96 | R18 |
| VSS_97 | R19 |
| VSS_98 | T8 |
| VSS_99 | T9 |
| VSS_100 | T10 |
| VSS_101 | T11 |
| VSS_102 | T12 |
| VSS_103 | T13 |
| VSS_104 | T14 |
| VSS_105 | T15 |
| VSS_106 | T16 |
| VSS_107 | T17 |
| VSS_108 | T18 |
| VSS_109 | T19 |
| VSS_110 | U6 |
| VSS_111 | U7 |
| VSS_112 | U8 |
| VSS_113 | U9 |
| VSS_114 | U10 |
| VSS_115 | U11 |
| VSS_116 | U12 |
| VSS_117 | U13 |
| VSS_118 | U14 |
| VSS_119 | U15 |
| VSS_120 | U16 |
| VSS_121 | U17 |
| VSS_122 | U18 |
| VSS_123 | U19 |
| VSS_124 | U20 |
| VSS_125 | U21 |
| VSS_126 | Y12 |
| VSS_127 | Y16 |
| VSS_128 | Y20 |
| VSS_129 | AB6 |
| VSS_130 | AB7 |
| VSS_131 | AB8 |
| VSS_132 | AB9 |
| VSS_133 | AB10 |
| VSS_134 | AB11 |
| VSS_135 | AB12 |
| VSS_136 | AB13 |
| VSS_137 | AB14 |
| VSS_138 | AB15 |
| VSS_139 | AB16 |
| VSS_140 | AB17 |
| VSS_141 | AB18 |
| VSS_142 | AB19 |
| VSS_143 | AB20 |
| VSS_144 | AB21 |
| VSS_145 | AA12 |
| VSS_146 | AA16 |
| VSS_147 | AA20 |
| VSS_148 | AC3 |
| VSS_149 | AD5 |
| VSS_150 | AD22 |
| VSS_151 | AE1 |
| VSS_152 | AE5 |
| VSS_153 | AE12 |
| VSS_154 | AE16 |
| VSS_155 | AE20 |
| VSS_156 | AE23 |
| VSS_157 | AE26 |
| VSS_158 | AF5 |
| VSS_159 | AF12 |
| VSS_160 | AF16 |
| VSS_161 | AF20 |
| VSS_162 | AF23 |
| VSS_163 | AE4 |

# 11 Pin Descriptions for VSC7426-02

The VSC7426-02 device has 672 pins, which are described in this section.

The pin information is also provided as an attached Microsoft Excel file, so that you can copy it electronically. In Adobe Reader, double-click the attachment icon.

## 11.1 Pin Diagram for VSC7426-02

The following illustration shows the pin diagram for the VSC7426-02 device. For clarity, the device is shown in two halves, the top left and top right.

***Figure 108*** • **Pin Diagram for VSC7426-02, Top Left**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | P8_D0P | P8_D1P | P8_D2P | P8_D3P | P7_D0P | P7_D1P | P7_D2P | P7_D3P | P6_D0P | P6_D1P | P6_D2P | P6_D3P |
| B | VSS_1 | P8_D0N | P8_D1N | P8_D2N | P8_D3N | P7_D0N | P7_D1N | P7_D2N | P7_D3N | P6_D0N | P6_D1N | P6_D2N | P6_D3N |
| C | P9_D3P | P9_D3N | COMA_MODE | nRESET | VDD_IO_21 | Reserved_1 | VCORE_CFG0 | VCORE_CFG1 | VCORE_CFG2 | VCore_ICE_nEn | Reserved_4 | RefClk_Sel0 | RefClk_Sel1 |
| D | P9_D2P | P9_D2N | Reserved_205 | VDD_AH_1 | VDD_AH_2 | Reserved_206 | Reserved_207 | Reserved_208 | Reserved_209 | Reserved_248 | VDD_AH_4 | Reserved_211 | Reserved_13 |
| E | P9_D1P | P9_D1N | Reserved_216 | VDD_AH_7 | VDD_AH_8 | VDD_IO_1 | VDD_IO_2 | VDD_AH_9 | VDD_AL_1 | VDD_AL_2 | VDD_AH_10 | VDD_AH_11 | Ref_rext_1 |
| F | P9_D0P | P9_D0N | Reserved_218 | VDD_AH_17 | VDD_AH_18 | VDD_IO_5 | VDD_AH_3 | VDD_AH_19 | VDD_AL_5 | VDD_AL_6 | VDD_AH_20 | VDD_AH_21 | Reserved_219 |
| G | P10_D3P | P10_D3N | VSS_3 | Reserved_15 | VSS_4 | VDD_1 | VDD_2 | VDD_3 | VDD_AL_9 | VDD_AL_10 | VDD_4 | VDD_5 | Reserved_247 |
| H | P10_D2P | P10_D2N | VSS_7 | Reserved_14 | VSS_8 | VDD_11 | VDD_12 | VDD_13 | VDD_14 | VDD_15 | VDD_16 | VDD_17 | Reserved_246 |
| J | P10_D1P | P10_D1N | VDD_AH_27 | VDD_AH_28 | VDD_AL_13 | VDD_AL_14 | VDD_AL_15 | Reserved_240 | Reserved_241 | Reserved_242 | Reserved_243 | Reserved_244 | Reserved_245 |
| K | P10_D0P | P10_D0N | VSS_11 | Ref_rext_2 | VDD_AL_19 | VDD_AL_20 | VDD_AL_21 | VSS_12 | VSS_13 | VSS_14 | VSS_15 | VSS_16 | VSS_17 |
| L | P11_D3P | P11_D3N | VSS_25 | Ref_filt_2 | VSS_26 | VDD_25 | VDD_26 | VSS_27 | VSS_28 | VSS_29 | VSS_30 | VSS_31 | VSS_32 |
| M | P11_D2P | P11_D2N | VDD_AH_31 | VDD_AH_32 | VDD_AH_33 | VDD_29 | VDD_30 | VSS_41 | VSS_42 | VSS_43 | VSS_44 | VSS_45 | VSS_46 |
| N | P11_D1P | P11_D1N | VSS_53 | VSS_54 | VSS_55 | VDD_33 | VDD_34 | VSS_56 | VSS_57 | VSS_58 | VSS_59 | VSS_60 | VSS_61 |
| P | P11_D0P | P11_D0N | VSS_71 | Reserved_24 | VDD_IO_7 | VDD_37 | VDD_38 | VSS_72 | VSS_73 | VSS_74 | VSS_75 | VSS_76 | VSS_77 |
| R | GPIO_31 | GPIO_30 | GPIO_29 | GPIO_28 | VDD_IO_8 | VDD_41 | VDD_42 | VSS_86 | VSS_87 | VSS_88 | VSS_89 | VSS_90 | VSS_91 |
| T | GPIO_27 | GPIO_26 | GPIO_25 | GPIO_24 | VDD_IO_9 | VDD_45 | VDD_46 | VSS_98 | VSS_99 | VSS_100 | VSS_101 | VSS_102 | VSS_103 |
| U | GPIO_23 | GPIO_22 | GPIO_21 | GPIO_20 | VDD_IO_10 | VSS_110 | VSS_111 | VSS_112 | VSS_113 | VSS_114 | VSS_115 | VSS_116 | VSS_117 |
| V | GPIO_19 | GPIO_18 | GPIO_17 | GPIO_16 | VDD_IO_11 | VDD_49 | VDD_50 | VDD_51 | VDD_52 | VDD_53 | VDD_54 | VDD_55 | VDD_56 |
| W | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | VDD_IO_12 | VDD_65 | VDD_66 | VDD_67 | VDD_68 | VDD_69 | VDD_70 | VDD_71 | VDD_72 |
| Y | GPIO_11 | GPIO_10 | GPIO_9 | GPIO_8 | VDD_IO_13 | Reserved_146 | Reserved_141 | RefClk_P | SerDes_E3_TxP | Reserved_134 | Reserved_129 | VSS_126 | SerDes_E2_TxP |
| AA | GPIO_7 | GPIO_6 | GPIO_5 | GPIO_4 | VDD_IO_14 | Reserved_147 | Reserved_140 | RefClk_N | SerDes_E3_TxN | Reserved_135 | Reserved_128 | VSS_145 | SerDes_E2_TxN |
| AB | GPIO_3 | GPIO_2 | GPIO_1 | GPIO_0 | VDD_IO_15 | VSS_129 | VSS_130 | VSS_131 | VSS_132 | VSS_133 | VSS_134 | VSS_135 | VSS_136 |
| AC | SI_DO | SI_nEn | VSS_148 | VDD_IO_16 | VDD_IO_17 | VDD_A_1 | VDD_A_2 | VDD_A_3 | VDD_A_4 | VDD_A_5 | VDD_A_6 | VDD_A_7 | VDD_A_8 |
| AD | SI_Clk | SI_DI | Reserved_18 | VDD_IO_18 | VSS_149 | VDD_VS_1 | VDD_VS_2 | VDD_VS_3 | VDD_VS_4 | VDD_VS_5 | VDD_VS_6 | VDD_VS_7 | VDD_VS_8 |
| AE | VSS_151 | Reserved_17 | VDD_IO_19 | VSS_163 | VSS_152 | Reserved_144 | Reserved_143 | Reserved_22 | SerDes_E3_RxP | Reserved_132 | Reserved_131 | VSS_153 | SerDes_E2_RxP |
| AF | | VDD_IO_20 | MDIO | MDC | VSS_158 | Reserved_145 | Reserved_142 | Reserved_23 | SerDes_E3_RxN | Reserved_133 | Reserved_130 | VSS_159 | SerDes_E2_RxN |

***Figure 109*** • **Pin Diagram for VSC7426-02, Top Right**

| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P5_D0P | P5_D1P | P5_D2P | P5_D3P | P4_D0P | P4_D1P | P4_D2P | P4_D3P | P3_D0P | P3_D1P | P3_D2P | P3_D3P | | A |
| P5_D0N | P5_D1N | P5_D2N | P5_D3N | P4_D0N | P4_D1N | P4_D2N | P4_D3N | P3_D0N | P3_D1N | P3_D2N | P3_D3N | VSS_2 | B |
| RefClk_Sel2 | Reserved_8 | Reserved_7 | Reserved_6 | Reserved_5 | Reserved_201 | Reserved_202 | Reserved_203 | THERMDC_VSS | THERMDA | Reserved_204 | P2_D0N | P2_D0P | C |
| Reserved_12 | Reserved_212 | VDD_AH_5 | JTAG_CLK | JTAG_DI | JTAG_DO | JTAG_TMS | JTAG_TRST | Reserved_213 | Reserved_214 | Reserved_215 | P2_D1N | P2_D1P | D |
| Ref_filt_1 | VDD_AH_12 | VDD_AH_13 | VDD_AL_3 | VDD_AL_4 | VDD_AH_14 | VDD_IO_3 | VDD_IO_4 | VDD_AH_15 | VDD_AH_16 | Reserved_217 | P2_D2N | P2_D2P | E |
| Reserved_220 | VDD_AH_22 | VDD_AH_23 | VDD_AL_7 | VDD_AL_8 | VDD_AH_24 | VDD_AH_6 | VDD_IO_6 | VDD_AH_25 | VDD_AH_26 | Reserved_221 | P2_D3N | P2_D3P | F |
| Reserved_223 | VDD_6 | VDD_7 | VDD_AL_11 | VDD_AL_12 | VDD_8 | VDD_9 | VDD_10 | VSS_5 | Reserved_10 | VSS_6 | P1_D0N | P1_D0P | G |
| Reserved_225 | VDD_18 | VDD_19 | VDD_20 | VDD_21 | VDD_22 | VDD_23 | VDD_24 | VSS_9 | Reserved_11 | VSS_10 | P1_D1N | P1_D1P | H |
| Reserved_232 | Reserved_233 | Reserved_234 | Reserved_235 | Reserved_236 | Reserved_237 | VDD_AL_16 | VDD_AL_17 | VDD_AL_18 | VDD_AH_29 | VDD_AH_30 | P1_D2N | P1_D2P | J |
| VSS_18 | VSS_19 | VSS_20 | VSS_21 | VSS_22 | VSS_23 | VDD_AL_22 | VDD_AL_23 | VDD_AL_24 | Ref_rext_0 | VSS_24 | P1_D3N | P1_D3P | K |
| VSS_33 | VSS_34 | VSS_35 | VSS_36 | VSS_37 | VSS_38 | VDD_27 | VDD_28 | VSS_39 | Ref_filt_0 | VSS_40 | P0_D0N | P0_D0P | L |
| VSS_47 | VSS_48 | VSS_49 | VSS_50 | VSS_51 | VSS_52 | VDD_31 | VDD_32 | VDD_AH_34 | VDD_AH_35 | VDD_AH_36 | P0_D1N | P0_D1P | M |
| VSS_62 | VSS_63 | VSS_64 | VSS_65 | VSS_66 | VSS_67 | VDD_35 | VDD_36 | VSS_68 | VSS_69 | VSS_70 | P0_D2N | P0_D2P | N |
| VSS_78 | VSS_79 | VSS_80 | VSS_81 | VSS_82 | VSS_83 | VDD_39 | VDD_40 | VDD_IODDR_1 | VSS_84 | VSS_85 | P0_D3N | P0_D3P | P |
| VSS_92 | VSS_93 | VSS_94 | VSS_95 | VSS_96 | VSS_97 | VDD_43 | VDD_44 | VDD_IODDR_2 | Reserved_20 | Reserved_19 | DDR_Rext | DDR_Vref | R |
| VSS_104 | VSS_105 | VSS_106 | VSS_107 | VSS_108 | VSS_109 | VDD_47 | VDD_48 | VDD_IODDR_3 | Reserved_21 | DDR_A13 | DDR_A12 | DDR_A11 | T |
| VSS_118 | VSS_119 | VSS_120 | VSS_121 | VSS_122 | VSS_123 | VSS_124 | VSS_125 | VDD_IODDR_4 | DDR_A7 | DDR_A9 | DDR_A6 | DDR_A8 | U |
| VDD_57 | VDD_58 | VDD_59 | VDD_60 | VDD_61 | VDD_62 | VDD_63 | VDD_64 | VDD_IODDR_5 | DDR_A3 | DDR_A5 | DDR_A2 | DDR_A4 | V |
| VDD_73 | VDD_74 | VDD_75 | VDD_76 | VDD_77 | VDD_78 | VDD_79 | VDD_80 | VDD_IODDR_6 | DDR_A10 | DDR_A1 | DDR_nCAS | DDR_A0 | W |
| Reserved_121 | Reserved_118 | VSS_127 | SerDes_E1_TxP | Reserved_110 | Reserved_105 | VSS_128 | Reserved_102 | VDD_IODDR_7 | DDR_BA0 | DDR_BA1 | DDR_ODT | DDR_nRAS | Y |
| Reserved_120 | Reserved_119 | VSS_146 | SerDes_E1_TxN | Reserved_111 | Reserved_104 | VSS_147 | Reserved_103 | VDD_IODDR_8 | DDR_nWE | DDR_BA2 | DDR_CK | DDR_CKn | AA |
| VSS_137 | VSS_138 | VSS_139 | VSS_140 | VSS_141 | VSS_142 | VSS_143 | VSS_144 | VDD_IODDR_9 | DDR_DQ3 | DDR_CKE | DDR_DQ2 | DDR_DQ5 | AB |
| VDD_A_9 | VDD_A_10 | VDD_A_11 | VDD_A_12 | VDD_A_13 | VDD_A_14 | VDD_A_15 | VDD_A_16 | VDD_IODDR_10 | DDR_DQ1 | DDR_DQ4 | DDR_DQ7 | DDR_DQ0 | AC |
| VDD_VS_9 | VDD_VS_10 | VDD_VS_11 | VDD_VS_12 | VDD_VS_13 | VDD_VS_14 | VDD_VS_15 | VDD_VS_16 | VSS_150 | VDD_IODDR_11 | DDR_DQ6 | DDR_DQS | DDR_DQSn | AD |
| Reserved_123 | Reserved_116 | VSS_154 | SerDes_E1_RxP | Reserved_108 | Reserved_107 | VSS_155 | Reserved_100 | SerDes_Rext_0 | VSS_156 | VDD_IODDR_12 | DDR_DM | VSS_157 | AE |
| Reserved_122 | Reserved_117 | VSS_160 | SerDes_E1_RxN | Reserved_109 | Reserved_106 | VSS_161 | Reserved_101 | SerDes_Rext_1 | VSS_162 | VDD_IODDR_14 | VDD_IODDR_13 | | AF |

# 11.2 Pins by Function for VSC7426-02

This section contains the functional pin descriptions for the VSC7426-02 device. The following table lists the definitions for the pin type symbols.

***Table 883*** • **Pin Type Symbol Definitions**

| Symbol | Pin Type | Description |
|---|---|---|
| ABIAS | Analog bias | Analog bias pin. |
| DIFF | Differential | Differential signal pair. |
| I | Input | Input signal. |
| O | Output | Output signal. |
| I/O | Bidirectional | Bidirectional input or output signal. |
| A | Analog input | Analog input for sensing variable voltage levels. |
| PD | Pull-down | On-chip pull-down resistor to VSS. |
| PU | Pull-up | On-chip pull-up resistor to VDD_IO. |
| 3V | | 3.3 V-tolerant. |
| O | Output | Output signal. |

*Table 883 •* **Pin Type Symbol Definitions *(continued)***

| Symbol | Pin Type | Description |
|---|---|---|
| OZ | 3-state output | Output. |
| ST | Schmitt-trigger | Input has Schmitt-trigger circuitry. |
| TD | Termination differential | Internal differential termination. |

## 11.2.1 Analog Bias Signals

The following table lists the pins associated with the analog bias signals.

*Table 884 •* **Analog Bias Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_Rext_[1:0] | A | Analog bias calibration.<br>Connect an external 620 Ω ±1% resistor between SerDes_Rext_1 and SerDes_Rext_0. |
| Ref_filt_[2:0] | A | Reference filter. Connect a 1.0 µF external capacitor between each pin and ground. |
| Ref_rext_[2:0] | A | Reference external resistor. Connect a 2.0 kΩ (1%) resistor between each pin and ground. |

## 11.2.2 Clock Circuits

The following table lists the pins associated with the system clock interface.

*Table 885 •* **System Clock Interface Pins**

| Name | Type | Description |
|---|---|---|
| RefClk_Sel[2:0] | I, PD | Reference clock frequency selection.<br>0: Connect to pull-down or leave floating.<br>1: Connect to pull-up to $V_{DD_IO}$.<br>Coding:<br>000: 125 MHz (default).<br>001: 156.25 MHz.<br>010: Reserved.<br>011: Reserved.<br>100: 25 MHz.<br>101: Reserved.<br>110: Reserved.<br>111: Reserved. |
| RefClk_P<br>RefClk_N | I, Diff | Reference clock input.<br>The input can be either differential or single-ended. In differential mode, REFCLK_P is the true part of the differential signal, and REFCLK_N is the complement part of the differential signal.<br>In single-ended mode, REFCLK_P is used as single-ended LVTTL input, and the REFCLK_N should be pulled to $V_{DD_A}$.<br>Required applied frequency depends on RefClk_Sel[2:0] input state. See description for RefClk_Sel[2:0] pins. |

### 11.2.3 DDR2 SDRAM Interface

The following table lists the pins associated with the DDR2 SDRAM interface.

***Table 886 •*** **DDR2 SDRAM Pins**

| Name | Type | Description |
|---|---|---|
| DDR_CK<br>DDR_CKn | 0, Diff | SDRAM differential clock. Differential clock to external SDRAM.<br>DDR_CK is the true part of the differential signal. DDR_nCk is the complement part. |
| DDR_CKE | O | SDRAM clock enable.<br>0: Disables clock in external SDRAM.<br>1: Enables clock in external SDRAM. |
| DDR_nRAS<br>DDR_nCAS<br>DDR_nWE | O | SDRAM command outputs.<br>DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered. |
| DDR_DM | O | SDRAM data mask outputs.<br>DDR_DM and DDR_UDM are mask signals for data written to the SDRAM. DDR_LDM corresponds to data on DDR_DQ[7:0], and DDR_UDM corresponds to data on DDR_DQ[15:8]. |
| DDR_BA[2:0] | O | SDRAM bank address outputs.<br>DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied. |
| DDR_A[13:0] | O | SDRAM address outputs.<br>Provide row and column addresses to the SDRAM. |
| DDR_DQ[7:0] | I/O | SDRAM data bus. |
| DDR_DQS<br>DDR_DQSn | I/O, Diff | SDRAM differential data strobes.<br>Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. |
| DDR_ODT | O | Control signals for the attached DDR2 SDRAM devices on-die termination. |
| DDR_Rext | ABIAS | External DDR impedance calibration.<br>Connect the pin through an external 240 Ω ±1% resistor to ground. |
| DDR_Vref | ABIAS | Input reference voltage.<br>Provides the input switching reference voltage to the SSTL DDR signals. |

### 11.2.4 General-Purpose Inputs and Outputs

The following table lists the pins associated with general-purpose inputs and outputs. The GPIO pins have an alternate function signal, which is mapped out in the following table. The overlaid functions are

selected by software on a pin-by-pin basis. The parallel interface and MIIM slave interface are enabled depending on the VCORE_CFG settings and override the normal GPIO and alternate functions.

***Table 887 •*** **GPIO Pin Mapping**

| Name | Overlaid Function 1 | Overlaid Function 2 | Parallel Interface | MIIM Slave Interface | Type |
|---|---|---|---|---|---|
| GPIO_0 | SIO_CLK | | | | I/O, PU,ST, 3V |
| GPIO_1 | SIO_LD | | | | I/O, PU,ST, 3V |
| GPIO_2 | SIO_DO | | | | I/O, PU,ST, 3V |
| GPIO_3 | SIO_DI | | | | I/O, PU,ST, 3V |
| GPIO_4 | TACHO | | | | I/O, PU,ST, 3V |
| GPIO_5 | TWI_SCL | PHY0_LED1 | | | I/O, PU,ST, 3V |
| GPIO_6 | TWI_SDA | PHY1_LED1 | | | I/O, PU,ST, 3V |
| GPIO_7 | | PHY2_LED1 | | | I/O, PU,ST, 3V |
| GPIO_8 | EXT_IRQ0 | PHY3_LED1 | | | I/O, PU,ST, 3V |
| GPIO_9 | EXT_IRQ1 | PHY4_LED1 | | | I/O, PU,ST, 3V |
| GPIO_10 | SFP14_SD | PHY5_LED1 | | | I/O, PU,ST, 3V |
| GPIO_11 | SFP15_SD | PHY6_LED1 | | | I/O, PU,ST, 3V |
| GPIO_12 | SFP17_SD | PHY7_LED1 | | | I/O, PU,ST, 3V |
| GPIO_13 | SFP18_SD | PHY8_LED1 | PI_nCS | | I/O, PU,ST, 3V |
| GPIO_14 | SI_nEN1 | PHY9_LED1 | PI_nWR | SLV_ADDR | I/O, PU,ST, 3V |
| GPIO_15 | SI_nEn2 | PHY10_LED1 | PI_nOE | SLV_MDC | I/O, PU,ST, 3V |
| GPIO_16 | SI_nEn3 | PHY11_LED1 | PI_nDone | SLV_MDIO | I/O, PU,ST, 3V |
| GPIO_17 | SFP10_SD | PHY0_LED0 | PI_A0 | | I/O, PU,ST, 3V |
| GPIO_18 | SFP11_SD | PHY2_LED0 | PI_A1 | | I/O, PU,ST, 3V |
| GPIO_19 | SFP12_SD | PHY2_LED0 | PI_A2 | | I/O, PU,ST, 3V |
| GPIO_20 | SFP13_SD | PHY3_LED0 | PI_A3 | | I/O, PU,ST, 3V |
| GPIO_21 | SFP16_SD | PHY4_LED0 | PI_D0 | | I/O, PU,ST, 3V |
| GPIO_22 | SFP19_SD | PHY5_LED0 | PI_D1 | | I/O, PU,ST, 3V |
| GPIO_23 | SFP24_SD | PHY6_LED0 | PI_D2 | | I/O, PU,ST, 3V |
| GPIO_24 | SFP25_SD | PHY7_LED0 | PI_D3 | | I/O, PU,ST, 3V |
| GPIO_25 | SFP20_SD | PHY8_LED0 | PI_D4 | | I/O, PU,ST, 3V |
| GPIO_26 | SFP21_SD | PHY9_LED0 | PI_D5 | | I/O, PU,ST, 3V |
| GPIO_27 | SFP22_SD | PHY10_LED0 | PI_D6 | | I/O, PU,ST, 3V |
| GPIO_28 | SFP23_SD | PHY11_LED0 | PI_D7 | | I/O, PU,ST, 3V |
| GPIO_29 | PWM | | | | I/O, PU,ST, 3V |
| GPIO_30 | UART_TX | | | | I/O, PU,ST, 3V |
| GPIO_31 | UART_RX | | | | I/O, PU,ST, 3V |

## 11.2.5 JTAG Interface

The following table lists the pins associated with the JTAG interface. The JTAG interface can be connected to the boundary scan TAP controller or the internal VCore-III TAP controller for software debug as described for the VCore_ICE_nEn signal.

The JTAG signals are not 5 V tolerant.

***Table 888 •*** **JTAG Interface Pins**

| Name | Type | Description |
|---|---|---|
| JTAG_nTRST | I, PU, ST, 3V | JTAG test reset, active low. For normal device operation, JTAG_nTRST should be pulled low. |
| JTAG_TCK | I, PU, ST, 3V | JTAG clock. |
| JTAG_TDI | I, PU, ST, 3V | JTAG test data in. |
| JTAG_TDO | OZ, 3V | JTAG test data out. |
| JTAG_TMS | I, PU, ST, 3V | JTAG test mode select. |

## 11.2.6 MII Management Interface

The following table lists the pins associated with the MII Management interface.

***Table 889 •*** **MII Management Interface Pins**

| Name | Type | Description |
|---|---|---|
| MDIO | I/O, 3V | Management data input/output.<br>MDIO is a bidirectional signal between a PHY and the device, used to transfer control and status information. Control information is driven by the device synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by the device. |
| MDC | O, 3V | Management data clock.<br>MDC is sourced by the station management entity (the device) to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal. |

## 11.2.7 Miscellaneous Signals

The following table lists the pins associated with a particular interface or facility on the device.

***Table 890 •*** **Miscellaneous Pins**

| Name | Type | Description |
|---|---|---|
| nReset | I, PD, ST, 3V | Global device reset, active low. |
| COMA_MODE | I/O, PU, ST, 3V | When this pin is asserted high, all PHYs are held in a powered-down state.<br>When this pin is deasserted low, all PHYs are powered up and resume normal operation. Additionally, this signal is used to synchronize the operation of multiple devices on the same printed circuit board to provide visual synchronization for LEDs driven from the separate devices. |

*Table 890 •* **Miscellaneous Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| VCORE_CFG[2:0] | I, PD, ST, 3V | Configuration signals for controlling the VCore-III CPU functions. |
| VCore_ICE_nEn | I, PU, 3V | VCore ICE nEn.<br>0: Enables the VCore-III JTAG debug interface over the JTAG interface pins.<br>1: Enables normal IO-JTAG over the JTAG interface. |
| THERMDA | A | Thermal diode anode (p-junction). |
| THERMDC_VSS | A | Thermal diode cathode (n-junction). Connected on-die to $V_{SS}$. |
| EXT_IRQ[1:0](1) | I/O PD, 3V | This pin interrupts inputs or outputs to the internal VCore-III CPU system or to an external processor. Signal polarity is programmable. |
| Reserved_1<br>Reserved_5<br>Reserved_7 | I, PD, ST, 3V | Tie to $V_{DD_IO}$. |
| Reserved_4<br>Reserved_6<br>Reserved_8 | I, PD, ST, 3V | Tie to $V_{SS}$. |
| Reserved_[10:15]<br>Reserved_[17:24]<br>Reserved_[100:111]<br>Reserved_[116:123]<br>Reserved_[128:135]<br>Reserved_[140:147]<br>Reserved_[201:209]<br>Reserved_[211:221]<br>Reserved_[223]<br>Reserved_[225]<br>Reserved_[232:237]<br>Reserved_[240:248] | I, PD, ST, 3V | Leave floating. |

1. Available as an alternate function on the GPIO_8 and GPIO_9 pins.

## 11.2.8 Parallel Interface

The parallel interface (PI) can operate in a Master mode or a Slave mode according to the VCORE_CFG[1:0] signal settings. In Master mode, the internal VCore-III CPU system controls the PI and can access external peripherals over it. In Slave mode, the PI can be used by an external CPU to access internal device resources.

The PI master and slave mode signals are alternate function signals on GPIO pins. For more information about the GPIO mapping, see Table 887, page 687.

*Table 891 •* **Parallel Interface VCore-III Master Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | OZ, 3V | External address bus.<br>Used for addressing external memory space. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | External data bus. PI_Data0 is LSB. |

*Table 891 •* **Parallel Interface VCore-III Master Mode Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| PI_nCS | OZ, 3V | Programmable active low chip selects.<br>PI_nCS is used as default for booting from external memory (typically Flash). |
| PI_nDone | I, 3V | Acknowledges an operation. Used for external device-paced access operation.<br>Signal polarity is programmable. |
| PI_nOE | OZ, 3V | Active low signal that signals external device to drive data bus during read access. |
| PI_nWR | OZ, 3V | Active low signal that signals external access direction. Read (1) or write (0). |

The following pins are associated with the parallel CPU interface slave mode.

*Table 892 •* **Parallel CPU Interface Slave Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | I, 3V | Internal device register address bus. Controlled by external CPU. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | Data bus.<br>PI_Data[0] is LSB. |
| PI_nCS | I, 3V | Device chip select. |
| PI_nDone | O, 3V | Acknowledges an operation. Signal polarity is programmable. |
| PI_nOE | I, 3V | Signals device to drive data bus during read operations. |
| PI_nWR | I, 3V | Signals access direction. Read (1) or write (0). |

## 11.2.9 Power Supplies and Ground

The following table lists the power supply and ground pins.

*Table 893 •* **Power Supply and Ground Pins**

| Name | Type | Description |
|---|---|---|
| VDD | Power | 1.0 V power supply voltage for core |
| VDD_A | Power | 1.0 V power supply voltage for analog circuits |
| VDD_AL | Power | 1.0 V power supply voltage for analog circuits for twisted pair interface |
| VDD_AH | Power | 2.5 V power supply voltage for analog driver in twisted pair interface |
| VDD_IO | Power | 2.5 V power supply for parallel CPU interface, MII Management interface, and miscellaneous I/Os |
| VDD_IODDR | Power | 1.8 V power supply for DDR interface |
| VDD_VS | Power | 1.0 V or 1.2 V power supply for SerDes and Enhanced SerDes interfaces |
| VSS | Ground | Ground reference |

## 11.2.10 Serial CPU Interface

The serial CPU interface (SI) can be used as a serial slave, master, or boot interface.

As a slave interface, it allows external CPUs to access internal registers. As a master interface, it allows the device to access external devices using a programmable protocol. The serial CPU interface also allows the internal VCore-III CPU system to boot from an attached serial memory device when the VCORE_CFG signals are set appropriately.

The following table lists the pins associated with the serial CPU interface (SI).

***Table 894 •*** **Serial CPU Interface Pins**

| Name | Type | Description |
|---|---|---|
| SI_Clk | I/O, 3V | Slave mode: Input receiving serial interface clock from external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven with clock to external serial memory device. |
| SI_DI | I, 3V | Slave mode: Input receiving serial interface data from external master.<br>Master mode: Input directly read by software from register bit.<br>Boot mode: Input boot data from external serial memory device. |
| SI_DO | OZ, 3V | Slave mode: Output transmitting serial interface data to external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: No function. |
| SI_nEn<br>SI_nEn[3:1](1) | I/O, 3V | Slave mode: Input used to enable SI slave interface.<br>0: Enabled<br>1: Disabled<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven while booting from EEPROM or serial flash to internal VCore-III CPU system. Released when booting is completed. |

1. Available as an alternate function on the GPIO_16, GPIO_15, and GPIO_14 pins. For more information about GPIO pin mapping, see Table 887, page 687.

## 11.2.11 Enhanced SerDes Interface

The following pins are associated with the Enhanced SerDes interface.

***Table 895 •*** **Enhanced SerDes Interface Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_E[3:1]_RxP, N | I, Diff, TD | Differential SGMII data inputs. |
| SerDes_E[3:1]_TxP, N | O, Diff | Differential SGMII data outputs. |

## 11.2.12 Twisted Pair Interface

The following pins are associated with the twisted pair interface. The PHYn_LED[1:0] LED control signals associated with the twisted pair interfaces are alternate functions on the GPIOs. For more information about the GPIO pin mapping, see Table 887, page 687.

***Table 896 •*** **Twisted Pair Interface Pins**

| Name | Type | Description |
|---|---|---|
| P0_D0P<br>P1_D0P<br>P2_D0P<br>P3_D0P<br>P4_D0P<br>P5_D0P<br>P6_D0P<br>P7_D0P<br>P8_D0P<br>P9_D0P<br>P10_D0P<br>P11_D0P | $A_{DIFF}$ | Tx/Rx channel A positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1. |
| P0_D0N<br>P1_D0N<br>P2_D0N<br>P3_D0N<br>P4_D0N<br>P5_D0N<br>P6_D0N<br>P7_D0N<br>P8_D0N<br>P9_D0N<br>P10_D0N<br>P11_D0N | $A_{DIFF}$ | Tx/Rx channel A negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2. |
| P0_D1P<br>P1_D1P<br>P2_D1P<br>P3_D1P<br>P4_D1P<br>P5_D1P<br>P6_D1P<br>P7_D1P<br>P8_D1P<br>P9_D1P<br>P10_D1P<br>P11_D1P | $A_{DIFF}$ | Tx/Rx channel B positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3. |
| P0_D1N<br>P1_D1N<br>P2_D1N<br>P3_D1N<br>P4_D1N<br>P5_D1N<br>P6_D1N<br>P7_D1N<br>P8_D1N<br>P9_D1N<br>P10_D1N<br>P11_D1N | $A_{DIFF}$ | Tx/Rx channel B negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6. |

*Table 896 •* **Twisted Pair Interface Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| P0_D2P<br>P1_D2P<br>P2_D2P<br>P3_D2P<br>P4_D2P<br>P5_D2P<br>P6_D2P<br>P7_D2P<br>P8_D2P<br>P9_D2P<br>P10_D2P<br>P11_D2P | $A_{DIFF}$ | Tx/Rx channel C positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes). |
| P0_D2N<br>P1_D2N<br>P2_D2N<br>P3_D2N<br>P4_D2N<br>P5_D2N<br>P6_D2N<br>P7_D2N<br>P8_D2N<br>P9_D2N<br>P10_D2N<br>P11_D2N | $A_{DIFF}$ | Tx/Rx channel C negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes). |
| P0_D3P<br>P1_D3P<br>P2_D3P<br>P3_D3P<br>P4_D3P<br>P5_D3P<br>P6_D3P<br>P7_D3P<br>P8_D3P<br>P9_D3P<br>P10_D3P<br>P11_D3P | $A_{DIFF}$ | Tx/Rx channel D positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes). |
| P0_D3N<br>P1_D3N<br>P2_D3N<br>P3_D3N<br>P4_D3N<br>P5_D3N<br>P6_D3N<br>P7_D3N<br>P8_D3N<br>P9_D3N<br>P10_D3N<br>P11_D3N | $A_{DIFF}$ | Tx/Rx channel D negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes). |

## 11.3 Pins by Number for VSC7426-02

This section provides a numeric list of the VSC7426-02 pins.

| | |
|---|---|
| A2 | P8_D0P |
| A3 | P8_D1P |
| A4 | P8_D2P |
| A5 | P8_D3P |
| A6 | P7_D0P |
| A7 | P7_D1P |
| A8 | P7_D2P |
| A9 | P7_D3P |
| A10 | P6_D0P |
| A11 | P6_D1P |
| A12 | P6_D2P |
| A13 | P6_D3P |
| A14 | P5_D0P |
| A15 | P5_D1P |
| A16 | P5_D2P |
| A17 | P5_D3P |
| A18 | P4_D0P |
| A19 | P4_D1P |
| A20 | P4_D2P |
| A21 | P4_D3P |
| A22 | P3_D0P |
| A23 | P3_D1P |
| A24 | P3_D2P |
| A25 | P3_D3P |
| AA1 | GPIO_7 |
| AA2 | GPIO_6 |
| AA3 | GPIO_5 |
| AA4 | GPIO_4 |
| AA5 | VDD_IO_14 |
| AA6 | Reserved_147 |
| AA7 | Reserved_140 |
| AA8 | RefClk_N |
| AA9 | SerDes_E3_TxN |
| AA10 | Reserved_135 |
| AA11 | Reserved_128 |
| AA12 | VSS_145 |
| AA13 | SerDes_E2_TxN |
| AA14 | Reserved_120 |
| AA15 | Reserved_119 |
| AA16 | VSS_146 |
| AA17 | SerDes_E1_TxN |
| AA18 | Reserved_111 |
| AA19 | Reserved_104 |
| AA20 | VSS_147 |
| AA21 | Reserved_103 |
| AA22 | VDD_IODDR_8 |
| AA23 | DDR_nWE |
| AA24 | DDR_BA2 |
| AA25 | DDR_CK |
| AA26 | DDR_CKn |
| AB1 | GPIO_3 |
| AB2 | GPIO_2 |
| AB3 | GPIO_1 |
| AB4 | GPIO_0 |
| AB5 | VDD_IO_15 |
| AB6 | VSS_129 |
| AB7 | VSS_130 |
| AB8 | VSS_131 |
| AB9 | VSS_132 |
| AB10 | VSS_133 |
| AB11 | VSS_134 |
| AB12 | VSS_135 |
| AB13 | VSS_136 |
| AB14 | VSS_137 |
| AB15 | VSS_138 |
| AB16 | VSS_139 |
| AB17 | VSS_140 |
| AB18 | VSS_141 |
| AB19 | VSS_142 |
| AB20 | VSS_143 |
| AB21 | VSS_144 |
| AB22 | VDD_IODDR_9 |
| AB23 | DDR_DQ3 |
| AB24 | DDR_CKE |
| AB25 | DDR_DQ2 |
| AB26 | DDR_DQ5 |
| AC1 | SI_DO |
| AC2 | SI_nEn |
| AC3 | VSS_148 |
| AC4 | VDD_IO_16 |
| AC5 | VDD_IO_17 |
| AC6 | VDD_A_1 |
| AC7 | VDD_A_2 |
| AC8 | VDD_A_3 |
| AC9 | VDD_A_4 |
| AC10 | VDD_A_5 |
| AC11 | VDD_A_6 |
| AC12 | VDD_A_7 |
| AC13 | VDD_A_8 |
| AC14 | VDD_A_9 |
| AC15 | VDD_A_10 |
| AC16 | VDD_A_11 |
| AC17 | VDD_A_12 |
| AC18 | VDD_A_13 |
| AC19 | VDD_A_14 |
| AC20 | VDD_A_15 |
| AC21 | VDD_A_16 |
| AC22 | VDD_IODDR_10 |
| AC23 | DDR_DQ1 |
| AC24 | DDR_DQ4 |
| AC25 | DDR_DQ7 |
| AC26 | DDR_DQ0 |
| AD1 | SI_Clk |
| AD2 | SI_DI |
| AD3 | Reserved_18 |
| AD4 | VDD_IO_18 |
| AD5 | VSS_149 |
| AD6 | VDD_VS_1 |
| AD7 | VDD_VS_2 |
| AD8 | VDD_VS_3 |

Pins by number *(continued)*

| | |
|---|---|
| AD9 | VDD_VS_4 |
| AD10 | VDD_VS_5 |
| AD11 | VDD_VS_6 |
| AD12 | VDD_VS_7 |
| AD13 | VDD_VS_8 |
| AD14 | VDD_VS_9 |
| AD15 | VDD_VS_10 |
| AD16 | VDD_VS_11 |
| AD17 | VDD_VS_12 |
| AD18 | VDD_VS_13 |
| AD19 | VDD_VS_14 |
| AD20 | VDD_VS_15 |
| AD21 | VDD_VS_16 |
| AD22 | VSS_150 |
| AD23 | VDD_IODDR_11 |
| AD24 | DDR_DQ6 |
| AD25 | DDR_DQS |
| AD26 | DDR_DQSn |
| AE1 | VSS_151 |
| AE2 | Reserved_17 |
| AE3 | VDD_IO_19 |
| AE4 | VSS_163 |
| AE5 | VSS_152 |
| AE6 | Reserved_144 |
| AE7 | Reserved_143 |
| AE8 | Reserved_22 |
| AE9 | SerDes_E3_RxP |
| AE10 | Reserved_132 |
| AE11 | Reserved_131 |
| AE12 | VSS_153 |
| AE13 | SerDes_E2_RxP |
| AE14 | Reserved_123 |
| AE15 | Reserved_116 |
| AE16 | VSS_154 |
| AE17 | SerDes_E1_RxP |
| AE18 | Reserved_108 |
| AE19 | Reserved_107 |
| AE20 | VSS_155 |
| AE21 | Reserved_100 |
| AE22 | SerDes_Rext_0 |
| AE23 | VSS_156 |
| AE24 | VDD_IODDR_12 |
| AE25 | DDR_DM |
| AE26 | VSS_157 |
| AF2 | VDD_IO_20 |
| AF3 | MDIO |
| AF4 | MDC |
| AF5 | VSS_158 |
| AF6 | Reserved_145 |
| AF7 | Reserved_142 |
| AF8 | Reserved_23 |
| AF9 | SerDes_E3_RxN |
| AF10 | Reserved_133 |
| AF11 | Reserved_130 |
| AF12 | VSS_159 |
| AF13 | SerDes_E2_RxN |
| AF14 | Reserved_122 |
| AF15 | Reserved_117 |
| AF16 | VSS_160 |
| AF17 | SerDes_E1_RxN |
| AF18 | Reserved_109 |
| AF19 | Reserved_106 |
| AF20 | VSS_161 |
| AF21 | Reserved_101 |
| AF22 | SerDes_Rext_1 |
| AF23 | VSS_162 |
| AF24 | VDD_IODDR_14 |
| AF25 | VDD_IODDR_13 |
| B1 | VSS_1 |
| B2 | P8_D0N |
| B3 | P8_D1N |
| B4 | P8_D2N |
| B5 | P8_D3N |
| B6 | P7_D0N |
| B7 | P7_D1N |
| B8 | P7_D2N |
| B9 | P7_D3N |
| B10 | P6_D0N |
| B11 | P6_D1N |
| B12 | P6_D2N |
| B13 | P6_D3N |
| B14 | P5_D0N |
| B15 | P5_D1N |
| B16 | P5_D2N |
| B17 | P5_D3N |
| B18 | P4_D0N |
| B19 | P4_D1N |
| B20 | P4_D2N |
| B21 | P4_D3N |
| B22 | P3_D0N |
| B23 | P3_D1N |
| B24 | P3_D2N |
| B25 | P3_D3N |
| B26 | VSS_2 |
| C1 | P9_D3P |
| C2 | P9_D3N |
| C3 | COMA_MODE |
| C4 | nRESET |
| C5 | VDD_IO_21 |
| C6 | Reserved_1 |
| C7 | VCORE_CFG0 |
| C8 | VCORE_CFG1 |
| C9 | VCORE_CFG2 |
| C10 | VCore_ICE_nEn |
| C11 | Reserved_4 |
| C12 | RefClk_Sel0 |
| C13 | RefClk_Sel1 |
| C14 | RefClk_Sel2 |
| C15 | Reserved_8 |
| C16 | Reserved_7 |
| C17 | Reserved_6 |
| C18 | Reserved_5 |
| C19 | Reserved_201 |
| C20 | Reserved_202 |
| C21 | Reserved_203 |
| C22 | THERMDC_VSS |
| C23 | THERMDA |

Pins by number *(continued)*

| | |
|---|---|
| C24 | Reserved_204 |
| C25 | P2_D0N |
| C26 | P2_D0P |
| D1 | P9_D2P |
| D2 | P9_D2N |
| D3 | Reserved_205 |
| D4 | VDD_AH_1 |
| D5 | VDD_AH_2 |
| D6 | Reserved_206 |
| D7 | Reserved_207 |
| D8 | Reserved_208 |
| D9 | Reserved_209 |
| D10 | Reserved_248 |
| D11 | VDD_AH_4 |
| D12 | Reserved_211 |
| D13 | Reserved_13 |
| D14 | Reserved_12 |
| D15 | Reserved_212 |
| D16 | VDD_AH_5 |
| D17 | JTAG_CLK |
| D18 | JTAG_DI |
| D19 | JTAG_DO |
| D20 | JTAG_TMS |
| D21 | JTAG_TRST |
| D22 | Reserved_213 |
| D23 | Reserved_214 |
| D24 | Reserved_215 |
| D25 | P2_D1N |
| D26 | P2_D1P |
| E1 | P9_D1P |
| E2 | P9_D1N |
| E3 | Reserved_216 |
| E4 | VDD_AH_7 |
| E5 | VDD_AH_8 |
| E6 | VDD_IO_1 |
| E7 | VDD_IO_2 |
| E8 | VDD_AH_9 |
| E9 | VDD_AL_1 |
| E10 | VDD_AL_2 |
| E11 | VDD_AH_10 |
| E12 | VDD_AH_11 |
| E13 | Ref_rext_1 |
| E14 | Ref_filt_1 |
| E15 | VDD_AH_12 |
| E16 | VDD_AH_13 |
| E17 | VDD_AL_3 |
| E18 | VDD_AL_4 |
| E19 | VDD_AH_14 |
| E20 | VDD_IO_3 |
| E21 | VDD_IO_4 |
| E22 | VDD_AH_15 |
| E23 | VDD_AH_16 |
| E24 | Reserved_217 |
| E25 | P2_D2N |
| E26 | P2_D2P |
| F1 | P9_D0P |
| F2 | P9_D0N |
| F3 | Reserved_218 |
| F4 | VDD_AH_17 |
| F5 | VDD_AH_18 |
| F6 | VDD_IO_5 |
| F7 | VDD_AH_3 |
| F8 | VDD_AH_19 |
| F9 | VDD_AL_5 |
| F10 | VDD_AL_6 |
| F11 | VDD_AH_20 |
| F12 | VDD_AH_21 |
| F13 | Reserved_219 |
| F14 | Reserved_220 |
| F15 | VDD_AH_22 |
| F16 | VDD_AH_23 |
| F17 | VDD_AL_7 |
| F18 | VDD_AL_8 |
| F19 | VDD_AH_24 |
| F20 | VDD_AH_6 |
| F21 | VDD_IO_6 |
| F22 | VDD_AH_25 |
| F23 | VDD_AH_26 |
| F24 | Reserved_221 |
| F25 | P2_D3N |
| F26 | P2_D3P |
| G1 | P10_D3P |
| G2 | P10_D3N |
| G3 | VSS_3 |
| G4 | Reserved_15 |
| G5 | VSS_4 |
| G6 | VDD_1 |
| G7 | VDD_2 |
| G8 | VDD_3 |
| G9 | VDD_AL_9 |
| G10 | VDD_AL_10 |
| G11 | VDD_4 |
| G12 | VDD_5 |
| G13 | Reserved_247 |
| G14 | Reserved_223 |
| G15 | VDD_6 |
| G16 | VDD_7 |
| G17 | VDD_AL_11 |
| G18 | VDD_AL_12 |
| G19 | VDD_8 |
| G20 | VDD_9 |
| G21 | VDD_10 |
| G22 | VSS_5 |
| G23 | Reserved_10 |
| G24 | VSS_6 |
| G25 | P1_D0N |
| G26 | P1_D0P |
| H1 | P10_D2P |
| H2 | P10_D2N |
| H3 | VSS_7 |
| H4 | Reserved_14 |
| H5 | VSS_8 |
| H6 | VDD_11 |
| H7 | VDD_12 |
| H8 | VDD_13 |
| H9 | VDD_14 |
| H10 | VDD_15 |

Pins by number *(continued)*

| | |
|---|---|
| H11 | VDD_16 |
| H12 | VDD_17 |
| H13 | Reserved_246 |
| H14 | Reserved_225 |
| H15 | VDD_18 |
| H16 | VDD_19 |
| H17 | VDD_20 |
| H18 | VDD_21 |
| H19 | VDD_22 |
| H20 | VDD_23 |
| H21 | VDD_24 |
| H22 | VSS_9 |
| H23 | Reserved_11 |
| H24 | VSS_10 |
| H25 | P1_D1N |
| H26 | P1_D1P |
| J1 | P10_D1P |
| J2 | P10_D1N |
| J3 | VDD_AH_27 |
| J4 | VDD_AH_28 |
| J5 | VDD_AL_13 |
| J6 | VDD_AL_14 |
| J7 | VDD_AL_15 |
| J8 | Reserved_240 |
| J9 | Reserved_241 |
| J10 | Reserved_242 |
| J11 | Reserved_243 |
| J12 | Reserved_244 |
| J13 | Reserved_245 |
| J14 | Reserved_232 |
| J15 | Reserved_233 |
| J16 | Reserved_234 |
| J17 | Reserved_235 |
| J18 | Reserved_236 |
| J19 | Reserved_237 |
| J20 | VDD_AL_16 |
| J21 | VDD_AL_17 |
| J22 | VDD_AL_18 |
| J23 | VDD_AH_29 |
| J24 | VDD_AH_30 |
| J25 | P1_D2N |
| J26 | P1_D2P |
| K1 | P10_D0P |
| K2 | P10_D0N |
| K3 | VSS_11 |
| K4 | Ref_rext_2 |
| K5 | VDD_AL_19 |
| K6 | VDD_AL_20 |
| K7 | VDD_AL_21 |
| K8 | VSS_12 |
| K9 | VSS_13 |
| K10 | VSS_14 |
| K11 | VSS_15 |
| K12 | VSS_16 |
| K13 | VSS_17 |
| K14 | VSS_18 |
| K15 | VSS_19 |
| K16 | VSS_20 |
| K17 | VSS_21 |
| K18 | VSS_22 |
| K19 | VSS_23 |
| K20 | VDD_AL_22 |
| K21 | VDD_AL_23 |
| K22 | VDD_AL_24 |
| K23 | Ref_rext_0 |
| K24 | VSS_24 |
| K25 | P1_D3N |
| K26 | P1_D3P |
| L1 | P11_D3P |
| L2 | P11_D3N |
| L3 | VSS_25 |
| L4 | Ref_filt_2 |
| L5 | VSS_26 |
| L6 | VDD_25 |
| L7 | VDD_26 |
| L8 | VSS_27 |
| L9 | VSS_28 |
| L10 | VSS_29 |
| L11 | VSS_30 |
| L12 | VSS_31 |
| L13 | VSS_32 |
| L14 | VSS_33 |
| L15 | VSS_34 |
| L16 | VSS_35 |
| L17 | VSS_36 |
| L18 | VSS_37 |
| L19 | VSS_38 |
| L20 | VDD_27 |
| L21 | VDD_28 |
| L22 | VSS_39 |
| L23 | Ref_filt_0 |
| L24 | VSS_40 |
| L25 | P0_D0N |
| L26 | P0_D0P |
| M1 | P11_D2P |
| M2 | P11_D2N |
| M3 | VDD_AH_31 |
| M4 | VDD_AH_32 |
| M5 | VDD_AH_33 |
| M6 | VDD_29 |
| M7 | VDD_30 |
| M8 | VSS_41 |
| M9 | VSS_42 |
| M10 | VSS_43 |
| M11 | VSS_44 |
| M12 | VSS_45 |
| M13 | VSS_46 |
| M14 | VSS_47 |
| M15 | VSS_48 |
| M16 | VSS_49 |
| M17 | VSS_50 |
| M18 | VSS_51 |
| M19 | VSS_52 |
| M20 | VDD_31 |
| M21 | VDD_32 |
| M22 | VDD_AH_34 |
| M23 | VDD_AH_35 |

Pins by number *(continued)*

| | |
|---|---|
| M24 | VDD_AH_36 |
| M25 | P0_D1N |
| M26 | P0_D1P |
| N1 | P11_D1P |
| N2 | P11_D1N |
| N3 | VSS_53 |
| N4 | VSS_54 |
| N5 | VSS_55 |
| N6 | VDD_33 |
| N7 | VDD_34 |
| N8 | VSS_56 |
| N9 | VSS_57 |
| N10 | VSS_58 |
| N11 | VSS_59 |
| N12 | VSS_60 |
| N13 | VSS_61 |
| N14 | VSS_62 |
| N15 | VSS_63 |
| N16 | VSS_64 |
| N17 | VSS_65 |
| N18 | VSS_66 |
| N19 | VSS_67 |
| N20 | VDD_35 |
| N21 | VDD_36 |
| N22 | VSS_68 |
| N23 | VSS_69 |
| N24 | VSS_70 |
| N25 | P0_D2N |
| N26 | P0_D2P |
| P1 | P11_D0P |
| P2 | P11_D0N |
| P3 | VSS_71 |
| P4 | Reserved_24 |
| P5 | VDD_IO_7 |
| P6 | VDD_37 |
| P7 | VDD_38 |
| P8 | VSS_72 |
| P9 | VSS_73 |
| P10 | VSS_74 |

| | |
|---|---|
| P11 | VSS_75 |
| P12 | VSS_76 |
| P13 | VSS_77 |
| P14 | VSS_78 |
| P15 | VSS_79 |
| P16 | VSS_80 |
| P17 | VSS_81 |
| P18 | VSS_82 |
| P19 | VSS_83 |
| P20 | VDD_39 |
| P21 | VDD_40 |
| P22 | VDD_IODDR_1 |
| P23 | VSS_84 |
| P24 | VSS_85 |
| P25 | P0_D3N |
| P26 | P0_D3P |
| R1 | GPIO_31 |
| R2 | GPIO_30 |
| R3 | GPIO_29 |
| R4 | GPIO_28 |
| R5 | VDD_IO_8 |
| R6 | VDD_41 |
| R7 | VDD_42 |
| R8 | VSS_86 |
| R9 | VSS_87 |
| R10 | VSS_88 |
| R11 | VSS_89 |
| R12 | VSS_90 |
| R13 | VSS_91 |
| R14 | VSS_92 |
| R15 | VSS_93 |
| R16 | VSS_94 |
| R17 | VSS_95 |
| R18 | VSS_96 |
| R19 | VSS_97 |
| R20 | VDD_43 |
| R21 | VDD_44 |
| R22 | VDD_IODDR_2 |
| R23 | Reserved_20 |

| | |
|---|---|
| R24 | Reserved_19 |
| R25 | DDR_Rext |
| R26 | DDR_Vref |
| T1 | GPIO_27 |
| T2 | GPIO_26 |
| T3 | GPIO_25 |
| T4 | GPIO_24 |
| T5 | VDD_IO_9 |
| T6 | VDD_45 |
| T7 | VDD_46 |
| T8 | VSS_98 |
| T9 | VSS_99 |
| T10 | VSS_100 |
| T11 | VSS_101 |
| T12 | VSS_102 |
| T13 | VSS_103 |
| T14 | VSS_104 |
| T15 | VSS_105 |
| T16 | VSS_106 |
| T17 | VSS_107 |
| T18 | VSS_108 |
| T19 | VSS_109 |
| T20 | VDD_47 |
| T21 | VDD_48 |
| T22 | VDD_IODDR_3 |
| T23 | Reserved_21 |
| T24 | DDR_A13 |
| T25 | DDR_A12 |
| T26 | DDR_A11 |
| U1 | GPIO_23 |
| U2 | GPIO_22 |
| U3 | GPIO_21 |
| U4 | GPIO_20 |
| U5 | VDD_IO_10 |
| U6 | VSS_110 |
| U7 | VSS_111 |
| U8 | VSS_112 |
| U9 | VSS_113 |
| U10 | VSS_114 |

Pins by number *(continued)*

| | |
|---|---|
| U11 | VSS_115 |
| U12 | VSS_116 |
| U13 | VSS_117 |
| U14 | VSS_118 |
| U15 | VSS_119 |
| U16 | VSS_120 |
| U17 | VSS_121 |
| U18 | VSS_122 |
| U19 | VSS_123 |
| U20 | VSS_124 |
| U21 | VSS_125 |
| U22 | VDD_IODDR_4 |
| U23 | DDR_A7 |
| U24 | DDR_A9 |
| U25 | DDR_A6 |
| U26 | DDR_A8 |
| V1 | GPIO_19 |
| V2 | GPIO_18 |
| V3 | GPIO_17 |
| V4 | GPIO_16 |
| V5 | VDD_IO_11 |
| V6 | VDD_49 |
| V7 | VDD_50 |
| V8 | VDD_51 |
| V9 | VDD_52 |
| V10 | VDD_53 |
| V11 | VDD_54 |
| V12 | VDD_55 |
| V13 | VDD_56 |
| V14 | VDD_57 |
| V15 | VDD_58 |
| V16 | VDD_59 |
| V17 | VDD_60 |
| V18 | VDD_61 |
| V19 | VDD_62 |
| V20 | VDD_63 |
| V21 | VDD_64 |
| V22 | VDD_IODDR_5 |
| V23 | DDR_A3 |
| V24 | DDR_A5 |
| V25 | DDR_A2 |
| V26 | DDR_A4 |
| W1 | GPIO_15 |
| W2 | GPIO_14 |
| W3 | GPIO_13 |
| W4 | GPIO_12 |
| W5 | VDD_IO_12 |
| W6 | VDD_65 |
| W7 | VDD_66 |
| W8 | VDD_67 |
| W9 | VDD_68 |
| W10 | VDD_69 |
| W11 | VDD_70 |
| W12 | VDD_71 |
| W13 | VDD_72 |
| W14 | VDD_73 |
| W15 | VDD_74 |
| W16 | VDD_75 |
| W17 | VDD_76 |
| W18 | VDD_77 |
| W19 | VDD_78 |
| W20 | VDD_79 |
| W21 | VDD_80 |
| W22 | VDD_IODDR_6 |
| W23 | DDR_A10 |
| W24 | DDR_A1 |
| W25 | DDR_nCAS |
| W26 | DDR_A0 |
| Y1 | GPIO_11 |
| Y2 | GPIO_10 |
| Y3 | GPIO_9 |
| Y4 | GPIO_8 |
| Y5 | VDD_IO_13 |
| Y6 | Reserved_146 |
| Y7 | Reserved_141 |
| Y8 | RefClk_P |
| Y9 | SerDes_E3_TxP |
| Y10 | Reserved_134 |
| Y11 | Reserved_129 |
| Y12 | VSS_126 |
| Y13 | SerDes_E2_TxP |
| Y14 | Reserved_121 |
| Y15 | Reserved_118 |
| Y16 | VSS_127 |
| Y17 | SerDes_E1_TxP |
| Y18 | Reserved_110 |
| Y19 | Reserved_105 |
| Y20 | VSS_128 |
| Y21 | Reserved_102 |
| Y22 | VDD_IODDR_7 |
| Y23 | DDR_BA0 |
| Y24 | DDR_BA1 |
| Y25 | DDR_ODT |
| Y26 | DDR_nRAS |

## 11.4 Pins by Name for VSC7426-02

This section provides an alphabetical list of the VSC7426-02 pins.

| | |
|---|---|
| COMA_MODE | C3 |
| DDR_A0 | W26 |
| DDR_A1 | W24 |
| DDR_A2 | V25 |
| DDR_A3 | V23 |
| DDR_A4 | V26 |
| DDR_A5 | V24 |
| DDR_A6 | U25 |
| DDR_A7 | U23 |
| DDR_A8 | U26 |
| DDR_A9 | U24 |
| DDR_A10 | W23 |
| DDR_A11 | T26 |
| DDR_A12 | T25 |
| DDR_A13 | T24 |
| DDR_BA0 | Y23 |
| DDR_BA1 | Y24 |
| DDR_BA2 | AA24 |
| DDR_CK | AA25 |
| DDR_CKE | AB24 |
| DDR_CKn | AA26 |
| DDR_DM | AE25 |
| DDR_DQ0 | AC26 |
| DDR_DQ1 | AC23 |
| DDR_DQ2 | AB25 |
| DDR_DQ3 | AB23 |
| DDR_DQ4 | AC24 |
| DDR_DQ5 | AB26 |
| DDR_DQ6 | AD24 |
| DDR_DQ7 | AC25 |
| DDR_DQS | AD25 |
| DDR_DQSn | AD26 |
| DDR_nCAS | W25 |
| DDR_nRAS | Y26 |
| DDR_nWE | AA23 |
| DDR_ODT | Y25 |
| DDR_Rext | R25 |
| DDR_Vref | R26 |
| GPIO_0 | AB4 |
| GPIO_1 | AB3 |
| GPIO_2 | AB2 |
| GPIO_3 | AB1 |
| GPIO_4 | AA4 |
| GPIO_5 | AA3 |
| GPIO_6 | AA2 |
| GPIO_7 | AA1 |
| GPIO_8 | Y4 |
| GPIO_9 | Y3 |
| GPIO_10 | Y2 |
| GPIO_11 | Y1 |
| GPIO_12 | W4 |
| GPIO_13 | W3 |
| GPIO_14 | W2 |
| GPIO_15 | W1 |
| GPIO_16 | V4 |
| GPIO_17 | V3 |
| GPIO_18 | V2 |
| GPIO_19 | V1 |
| GPIO_20 | U4 |
| GPIO_21 | U3 |
| GPIO_22 | U2 |
| GPIO_23 | U1 |
| GPIO_24 | T4 |
| GPIO_25 | T3 |
| GPIO_26 | T2 |
| GPIO_27 | T1 |
| GPIO_28 | R4 |
| GPIO_29 | R3 |
| GPIO_30 | R2 |
| GPIO_31 | R1 |
| JTAG_CLK | D17 |
| JTAG_DI | D18 |
| JTAG_DO | D19 |
| JTAG_TMS | D20 |
| JTAG_TRST | D21 |
| MDC | AF4 |
| MDIO | AF3 |
| nRESET | C4 |
| P0_D0N | L25 |
| P0_D0P | L26 |
| P0_D1N | M25 |
| P0_D1P | M26 |
| P0_D2N | N25 |
| P0_D2P | N26 |
| P0_D3N | P25 |
| P0_D3P | P26 |
| P1_D0N | G25 |
| P1_D0P | G26 |
| P1_D1N | H25 |
| P1_D1P | H26 |
| P1_D2N | J25 |
| P1_D2P | J26 |
| P1_D3N | K25 |
| P1_D3P | K26 |
| P2_D0N | C25 |
| P2_D0P | C26 |
| P2_D1N | D25 |
| P2_D1P | D26 |
| P2_D2N | E25 |
| P2_D2P | E26 |
| P2_D3N | F25 |
| P2_D3P | F26 |
| P3_D0N | B22 |
| P3_D0P | A22 |
| P3_D1N | B23 |
| P3_D1P | A23 |
| P3_D2N | B24 |
| P3_D2P | A24 |
| P3_D3N | B25 |
| P3_D3P | A25 |

Pins by name *(continued)*

| | |
|---|---|
| P4_D0N | B18 |
| P4_D0P | A18 |
| P4_D1N | B19 |
| P4_D1P | A19 |
| P4_D2N | B20 |
| P4_D2P | A20 |
| P4_D3N | B21 |
| P4_D3P | A21 |
| P5_D0N | B14 |
| P5_D0P | A14 |
| P5_D1N | B15 |
| P5_D1P | A15 |
| P5_D2N | B16 |
| P5_D2P | A16 |
| P5_D3N | B17 |
| P5_D3P | A17 |
| P6_D0N | B10 |
| P6_D0P | A10 |
| P6_D1N | B11 |
| P6_D1P | A11 |
| P6_D2N | B12 |
| P6_D2P | A12 |
| P6_D3N | B13 |
| P6_D3P | A13 |
| P7_D0N | B6 |
| P7_D0P | A6 |
| P7_D1N | B7 |
| P7_D1P | A7 |
| P7_D2N | B8 |
| P7_D2P | A8 |
| P7_D3N | B9 |
| P7_D3P | A9 |
| P8_D0N | B2 |
| P8_D0P | A2 |
| P8_D1N | B3 |
| P8_D1P | A3 |
| P8_D2N | B4 |
| P8_D2P | A4 |
| P8_D3N | B5 |
| P8_D3P | A5 |
| P9_D0N | F2 |
| P9_D0P | F1 |
| P9_D1N | E2 |
| P9_D1P | E1 |
| P9_D2N | D2 |
| P9_D2P | D1 |
| P9_D3N | C2 |
| P9_D3P | C1 |
| P10_D0N | K2 |
| P10_D0P | K1 |
| P10_D1N | J2 |
| P10_D1P | J1 |
| P10_D2N | H2 |
| P10_D2P | H1 |
| P10_D3N | G2 |
| P10_D3P | G1 |
| P11_D0N | P2 |
| P11_D0P | P1 |
| P11_D1N | N2 |
| P11_D1P | N1 |
| P11_D2N | M2 |
| P11_D2P | M1 |
| P11_D3N | L2 |
| P11_D3P | L1 |
| Ref_filt_0 | L23 |
| Ref_filt_1 | E14 |
| Ref_filt_2 | L4 |
| Ref_rext_0 | K23 |
| Ref_rext_1 | E13 |
| Ref_rext_2 | K4 |
| RefClk_N | AA8 |
| RefClk_P | Y8 |
| RefClk_Sel0 | C12 |
| RefClk_Sel1 | C13 |
| RefClk_Sel2 | C14 |
| Reserved_1 | C6 |
| Reserved_4 | C11 |
| Reserved_5 | C18 |
| Reserved_6 | C17 |
| Reserved_7 | C16 |
| Reserved_8 | C15 |
| Reserved_10 | G23 |
| Reserved_11 | H23 |
| Reserved_12 | D14 |
| Reserved_13 | D13 |
| Reserved_14 | H4 |
| Reserved_15 | G4 |
| Reserved_17 | AE2 |
| Reserved_18 | AD3 |
| Reserved_19 | R24 |
| Reserved_20 | R23 |
| Reserved_21 | T23 |
| Reserved_22 | AE8 |
| Reserved_23 | AF8 |
| Reserved_24 | P4 |
| Reserved_100 | AE21 |
| Reserved_101 | AF21 |
| Reserved_102 | Y21 |
| Reserved_103 | AA21 |
| Reserved_104 | AA19 |
| Reserved_105 | Y19 |
| Reserved_106 | AF19 |
| Reserved_107 | AE19 |
| Reserved_108 | AE18 |
| Reserved_109 | AF18 |
| Reserved_110 | Y18 |
| Reserved_111 | AA18 |
| Reserved_116 | AE15 |
| Reserved_117 | AF15 |
| Reserved_118 | Y15 |
| Reserved_119 | AA15 |
| Reserved_120 | AA14 |
| Reserved_121 | Y14 |
| Reserved_122 | AF14 |
| Reserved_123 | AE14 |
| Reserved_128 | AA11 |
| Reserved_129 | Y11 |

Pins by name *(continued)*

| | |
|---|---|
| Reserved_130 | AF11 |
| Reserved_131 | AE11 |
| Reserved_132 | AE10 |
| Reserved_133 | AF10 |
| Reserved_134 | Y10 |
| Reserved_135 | AA10 |
| Reserved_140 | AA7 |
| Reserved_141 | Y7 |
| Reserved_142 | AF7 |
| Reserved_143 | AE7 |
| Reserved_144 | AE6 |
| Reserved_145 | AF6 |
| Reserved_146 | Y6 |
| Reserved_147 | AA6 |
| Reserved_201 | C19 |
| Reserved_202 | C20 |
| Reserved_203 | C21 |
| Reserved_204 | C24 |
| Reserved_205 | D3 |
| Reserved_206 | D6 |
| Reserved_207 | D7 |
| Reserved_208 | D8 |
| Reserved_209 | D9 |
| Reserved_211 | D12 |
| Reserved_212 | D15 |
| Reserved_213 | D22 |
| Reserved_214 | D23 |
| Reserved_215 | D24 |
| Reserved_216 | E3 |
| Reserved_217 | E24 |
| Reserved_218 | F3 |
| Reserved_219 | F13 |
| Reserved_220 | F14 |
| Reserved_221 | F24 |
| Reserved_223 | G14 |
| Reserved_225 | H14 |
| Reserved_232 | J14 |
| Reserved_233 | J15 |
| Reserved_234 | J16 |
| Reserved_235 | J17 |
| Reserved_236 | J18 |
| Reserved_237 | J19 |
| Reserved_240 | J8 |
| Reserved_241 | J9 |
| Reserved_242 | J10 |
| Reserved_243 | J11 |
| Reserved_244 | J12 |
| Reserved_245 | J13 |
| Reserved_246 | H13 |
| Reserved_247 | G13 |
| Reserved_248 | D10 |
| SerDes_E1_RxN | AF17 |
| SerDes_E1_RxP | AE17 |
| SerDes_E1_TxN | AA17 |
| SerDes_E1_TxP | Y17 |
| SerDes_E2_RxN | AF13 |
| SerDes_E2_RxP | AE13 |
| SerDes_E2_TxN | AA13 |
| SerDes_E2_TxP | Y13 |
| SerDes_E3_RxN | AF9 |
| SerDes_E3_RxP | AE9 |
| SerDes_E3_TxN | AA9 |
| SerDes_E3_TxP | Y9 |
| SerDes_Rext_0 | AE22 |
| SerDes_Rext_1 | AF22 |
| SI_Clk | AD1 |
| SI_DI | AD2 |
| SI_DO | AC1 |
| SI_nEn | AC2 |
| THERMDA | C23 |
| THERMDC_VSS | C22 |
| VCORE_CFG0 | C7 |
| VCORE_CFG1 | C8 |
| VCORE_CFG2 | C9 |
| VCore_ICE_nEn | C10 |
| VDD_1 | G6 |
| VDD_2 | G7 |
| VDD_3 | G8 |
| VDD_4 | G11 |
| VDD_5 | G12 |
| VDD_6 | G15 |
| VDD_7 | G16 |
| VDD_8 | G19 |
| VDD_9 | G20 |
| VDD_10 | G21 |
| VDD_11 | H6 |
| VDD_12 | H7 |
| VDD_13 | H8 |
| VDD_14 | H9 |
| VDD_15 | H10 |
| VDD_16 | H11 |
| VDD_17 | H12 |
| VDD_18 | H15 |
| VDD_19 | H16 |
| VDD_20 | H17 |
| VDD_21 | H18 |
| VDD_22 | H19 |
| VDD_23 | H20 |
| VDD_24 | H21 |
| VDD_25 | L6 |
| VDD_26 | L7 |
| VDD_27 | L20 |
| VDD_28 | L21 |
| VDD_29 | M6 |
| VDD_30 | M7 |
| VDD_31 | M20 |
| VDD_32 | M21 |
| VDD_33 | N6 |
| VDD_34 | N7 |
| VDD_35 | N20 |
| VDD_36 | N21 |
| VDD_37 | P6 |
| VDD_38 | P7 |
| VDD_39 | P20 |
| VDD_40 | P21 |
| VDD_41 | R6 |
| VDD_42 | R7 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_43 | R20 |
| VDD_44 | R21 |
| VDD_45 | T6 |
| VDD_46 | T7 |
| VDD_47 | T20 |
| VDD_48 | T21 |
| VDD_49 | V6 |
| VDD_50 | V7 |
| VDD_51 | V8 |
| VDD_52 | V9 |
| VDD_53 | V10 |
| VDD_54 | V11 |
| VDD_55 | V12 |
| VDD_56 | V13 |
| VDD_57 | V14 |
| VDD_58 | V15 |
| VDD_59 | V16 |
| VDD_60 | V17 |
| VDD_61 | V18 |
| VDD_62 | V19 |
| VDD_63 | V20 |
| VDD_64 | V21 |
| VDD_65 | W6 |
| VDD_66 | W7 |
| VDD_67 | W8 |
| VDD_68 | W9 |
| VDD_69 | W10 |
| VDD_70 | W11 |
| VDD_71 | W12 |
| VDD_72 | W13 |
| VDD_73 | W14 |
| VDD_74 | W15 |
| VDD_75 | W16 |
| VDD_76 | W17 |
| VDD_77 | W18 |
| VDD_78 | W19 |
| VDD_79 | W20 |
| VDD_80 | W21 |
| VDD_A_1 | AC6 |
| VDD_A_2 | AC7 |
| VDD_A_3 | AC8 |
| VDD_A_4 | AC9 |
| VDD_A_5 | AC10 |
| VDD_A_6 | AC11 |
| VDD_A_7 | AC12 |
| VDD_A_8 | AC13 |
| VDD_A_9 | AC14 |
| VDD_A_10 | AC15 |
| VDD_A_11 | AC16 |
| VDD_A_12 | AC17 |
| VDD_A_13 | AC18 |
| VDD_A_14 | AC19 |
| VDD_A_15 | AC20 |
| VDD_A_16 | AC21 |
| VDD_AH_1 | D4 |
| VDD_AH_2 | D5 |
| VDD_AH_3 | F7 |
| VDD_AH_4 | D11 |
| VDD_AH_5 | D16 |
| VDD_AH_6 | F20 |
| VDD_AH_7 | E4 |
| VDD_AH_8 | E5 |
| VDD_AH_9 | E8 |
| VDD_AH_10 | E11 |
| VDD_AH_11 | E12 |
| VDD_AH_12 | E15 |
| VDD_AH_13 | E16 |
| VDD_AH_14 | E19 |
| VDD_AH_15 | E22 |
| VDD_AH_16 | E23 |
| VDD_AH_17 | F4 |
| VDD_AH_18 | F5 |
| VDD_AH_19 | F8 |
| VDD_AH_20 | F11 |
| VDD_AH_21 | F12 |
| VDD_AH_22 | F15 |
| VDD_AH_23 | F16 |
| VDD_AH_24 | F19 |
| VDD_AH_25 | F22 |
| VDD_AH_26 | F23 |
| VDD_AH_27 | J3 |
| VDD_AH_28 | J4 |
| VDD_AH_29 | J23 |
| VDD_AH_30 | J24 |
| VDD_AH_31 | M3 |
| VDD_AH_32 | M4 |
| VDD_AH_33 | M5 |
| VDD_AH_34 | M22 |
| VDD_AH_35 | M23 |
| VDD_AH_36 | M24 |
| VDD_AL_1 | E9 |
| VDD_AL_2 | E10 |
| VDD_AL_3 | E17 |
| VDD_AL_4 | E18 |
| VDD_AL_5 | F9 |
| VDD_AL_6 | F10 |
| VDD_AL_7 | F17 |
| VDD_AL_8 | F18 |
| VDD_AL_9 | G9 |
| VDD_AL_10 | G10 |
| VDD_AL_11 | G17 |
| VDD_AL_12 | G18 |
| VDD_AL_13 | J5 |
| VDD_AL_14 | J6 |
| VDD_AL_15 | J7 |
| VDD_AL_16 | J20 |
| VDD_AL_17 | J21 |
| VDD_AL_18 | J22 |
| VDD_AL_19 | K5 |
| VDD_AL_20 | K6 |
| VDD_AL_21 | K7 |
| VDD_AL_22 | K20 |
| VDD_AL_23 | K21 |
| VDD_AL_24 | K22 |
| VDD_IO_1 | E6 |
| VDD_IO_2 | E7 |
| VDD_IO_3 | E20 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_IO_4 | E21 |
| VDD_IO_5 | F6 |
| VDD_IO_6 | F21 |
| VDD_IO_7 | P5 |
| VDD_IO_8 | R5 |
| VDD_IO_9 | T5 |
| VDD_IO_10 | U5 |
| VDD_IO_11 | V5 |
| VDD_IO_12 | W5 |
| VDD_IO_13 | Y5 |
| VDD_IO_14 | AA5 |
| VDD_IO_15 | AB5 |
| VDD_IO_16 | AC4 |
| VDD_IO_17 | AC5 |
| VDD_IO_18 | AD4 |
| VDD_IO_19 | AE3 |
| VDD_IO_20 | AF2 |
| VDD_IO_21 | C5 |
| VDD_IODDR_1 | P22 |
| VDD_IODDR_2 | R22 |
| VDD_IODDR_3 | T22 |
| VDD_IODDR_4 | U22 |
| VDD_IODDR_5 | V22 |
| VDD_IODDR_6 | W22 |
| VDD_IODDR_7 | Y22 |
| VDD_IODDR_8 | AA22 |
| VDD_IODDR_9 | AB22 |
| VDD_IODDR_10 | AC22 |
| VDD_IODDR_11 | AD23 |
| VDD_IODDR_12 | AE24 |
| VDD_IODDR_13 | AF25 |
| VDD_IODDR_14 | AF24 |
| VDD_VS_1 | AD6 |
| VDD_VS_2 | AD7 |
| VDD_VS_3 | AD8 |
| VDD_VS_4 | AD9 |
| VDD_VS_5 | AD10 |
| VDD_VS_6 | AD11 |
| VDD_VS_7 | AD12 |
| VDD_VS_8 | AD13 |
| VDD_VS_9 | AD14 |
| VDD_VS_10 | AD15 |
| VDD_VS_11 | AD16 |
| VDD_VS_12 | AD17 |
| VDD_VS_13 | AD18 |
| VDD_VS_14 | AD19 |
| VDD_VS_15 | AD20 |
| VDD_VS_16 | AD21 |
| VSS_1 | B1 |
| VSS_2 | B26 |
| VSS_3 | G3 |
| VSS_4 | G5 |
| VSS_5 | G22 |
| VSS_6 | G24 |
| VSS_7 | H3 |
| VSS_8 | H5 |
| VSS_9 | H22 |
| VSS_10 | H24 |
| VSS_11 | K3 |
| VSS_12 | K8 |
| VSS_13 | K9 |
| VSS_14 | K10 |
| VSS_15 | K11 |
| VSS_16 | K12 |
| VSS_17 | K13 |
| VSS_18 | K14 |
| VSS_19 | K15 |
| VSS_20 | K16 |
| VSS_21 | K17 |
| VSS_22 | K18 |
| VSS_23 | K19 |
| VSS_24 | K24 |
| VSS_25 | L3 |
| VSS_26 | L5 |
| VSS_27 | L8 |
| VSS_28 | L9 |
| VSS_29 | L10 |
| VSS_30 | L11 |
| VSS_31 | L12 |
| VSS_32 | L13 |
| VSS_33 | L14 |
| VSS_34 | L15 |
| VSS_35 | L16 |
| VSS_36 | L17 |
| VSS_37 | L18 |
| VSS_38 | L19 |
| VSS_39 | L22 |
| VSS_40 | L24 |
| VSS_41 | M8 |
| VSS_42 | M9 |
| VSS_43 | M10 |
| VSS_44 | M11 |
| VSS_45 | M12 |
| VSS_46 | M13 |
| VSS_47 | M14 |
| VSS_48 | M15 |
| VSS_49 | M16 |
| VSS_50 | M17 |
| VSS_51 | M18 |
| VSS_52 | M19 |
| VSS_53 | N3 |
| VSS_54 | N4 |
| VSS_55 | N5 |
| VSS_56 | N8 |
| VSS_57 | N9 |
| VSS_58 | N10 |
| VSS_59 | N11 |
| VSS_60 | N12 |
| VSS_61 | N13 |
| VSS_62 | N14 |
| VSS_63 | N15 |
| VSS_64 | N16 |
| VSS_65 | N17 |
| VSS_66 | N18 |
| VSS_67 | N19 |
| VSS_68 | N22 |
| VSS_69 | N23 |

Pins by name *(continued)*

| | |
|---|---|
| VSS_70 | N24 |
| VSS_71 | P3 |
| VSS_72 | P8 |
| VSS_73 | P9 |
| VSS_74 | P10 |
| VSS_75 | P11 |
| VSS_76 | P12 |
| VSS_77 | P13 |
| VSS_78 | P14 |
| VSS_79 | P15 |
| VSS_80 | P16 |
| VSS_81 | P17 |
| VSS_82 | P18 |
| VSS_83 | P19 |
| VSS_84 | P23 |
| VSS_85 | P24 |
| VSS_86 | R8 |
| VSS_87 | R9 |
| VSS_88 | R10 |
| VSS_89 | R11 |
| VSS_90 | R12 |
| VSS_91 | R13 |
| VSS_92 | R14 |
| VSS_93 | R15 |
| VSS_94 | R16 |
| VSS_95 | R17 |
| VSS_96 | R18 |
| VSS_97 | R19 |
| VSS_98 | T8 |
| VSS_99 | T9 |
| VSS_100 | T10 |
| VSS_101 | T11 |
| VSS_102 | T12 |
| VSS_103 | T13 |
| VSS_104 | T14 |
| VSS_105 | T15 |
| VSS_106 | T16 |
| VSS_107 | T17 |
| VSS_108 | T18 |
| VSS_109 | T19 |
| VSS_110 | U6 |
| VSS_111 | U7 |
| VSS_112 | U8 |
| VSS_113 | U9 |
| VSS_114 | U10 |
| VSS_115 | U11 |
| VSS_116 | U12 |
| VSS_117 | U13 |
| VSS_118 | U14 |
| VSS_119 | U15 |
| VSS_120 | U16 |
| VSS_121 | U17 |
| VSS_122 | U18 |
| VSS_123 | U19 |
| VSS_124 | U20 |
| VSS_125 | U21 |
| VSS_126 | Y12 |
| VSS_127 | Y16 |
| VSS_128 | Y20 |
| VSS_129 | AB6 |
| VSS_130 | AB7 |
| VSS_131 | AB8 |
| VSS_132 | AB9 |
| VSS_133 | AB10 |
| VSS_134 | AB11 |
| VSS_135 | AB12 |
| VSS_136 | AB13 |
| VSS_137 | AB14 |
| VSS_138 | AB15 |
| VSS_139 | AB16 |
| VSS_140 | AB17 |
| VSS_141 | AB18 |
| VSS_142 | AB19 |
| VSS_143 | AB20 |
| VSS_144 | AB21 |
| VSS_145 | AA12 |
| VSS_146 | AA16 |
| VSS_147 | AA20 |
| VSS_148 | AC3 |
| VSS_149 | AD5 |
| VSS_150 | AD22 |
| VSS_151 | AE1 |
| VSS_152 | AE5 |
| VSS_153 | AE12 |
| VSS_154 | AE16 |
| VSS_155 | AE20 |
| VSS_156 | AE23 |
| VSS_157 | AE26 |
| VSS_158 | AF5 |
| VSS_159 | AF12 |
| VSS_160 | AF16 |
| VSS_161 | AF20 |
| VSS_162 | AF23 |
| VSS_163 | AE4 |

# 12 Pin Descriptions for VSC7427-02

The VSC7427-02 device has 672 pins, which are described in this section.

The pin information is also provided as an attached Microsoft Excel file, so that you can copy it electronically. In Adobe Reader, double-click the attachment icon.

## 12.1 Pin Diagram for VSC7427-02

The following illustration shows the pin diagram for the VSC7427-02 device. For clarity, the device is shown in two halves, the top left and top right.

***Figure 110*** • **Pin Diagram for VSC7427-02, Top Left**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | P8_D0P | P8_D1P | P8_D2P | P8_D3P | P7_D0P | P7_D1P | P7_D2P | P7_D3P | P6_D0P | P6_D1P | P6_D2P | P6_D3P |
| B | VSS_1 | P8_D0N | P8_D1N | P8_D2N | P8_D3N | P7_D0N | P7_D1N | P7_D2N | P7_D3N | P6_D0N | P6_D1N | P6_D2N | P6_D3N |
| C | P9_D3P | P9_D3N | COMA_MODE | nRESET | VDD_IO_21 | Reserved_1 | VCORE_CFG0 | VCORE_CFG1 | VCORE_CFG2 | VCore_ICE_nEn | Reserved_4 | RefClk_Sel0 | RefClk_Sel1 |
| D | P9_D2P | P9_D2N | Reserved_205 | VDD_AH_1 | VDD_AH_2 | Reserved_206 | Reserved_207 | Reserved_208 | Reserved_209 | Reserved_248 | VDD_AH_4 | Reserved_211 | Reserved_13 |
| E | P9_D1P | P9_D1N | Reserved_216 | VDD_AH_7 | VDD_AH_8 | VDD_IO_1 | VDD_IO_2 | VDD_AH_9 | VDD_AL_1 | VDD_AL_2 | VDD_AH_10 | VDD_AH_11 | Ref_rext_1 |
| F | P9_D0P | P9_D0N | Reserved_218 | VDD_AH_17 | VDD_AH_18 | VDD_IO_5 | VDD_AH_3 | VDD_AH_19 | VDD_AL_5 | VDD_AL_6 | VDD_AH_20 | VDD_AH_21 | Reserved_219 |
| G | P10_D3P | P10_D3N | VSS_3 | Reserved_15 | VSS_4 | VDD_1 | VDD_2 | VDD_3 | VDD_AL_9 | VDD_AL_10 | VDD_4 | VDD_5 | Reserved_247 |
| H | P10_D2P | P10_D2N | VSS_7 | Reserved_14 | VSS_8 | VDD_11 | VDD_12 | VDD_13 | VDD_14 | VDD_15 | VDD_16 | VDD_17 | Reserved_246 |
| J | P10_D1P | P10_D1N | VDD_AH_27 | VDD_AH_28 | VDD_AL_13 | VDD_AL_14 | VDD_AL_15 | Reserved_240 | Reserved_241 | Reserved_242 | Reserved_243 | Reserved_244 | Reserved_245 |
| K | P10_D0P | P10_D0N | VSS_11 | Ref_rext_2 | VDD_AL_19 | VDD_AL_20 | VDD_AL_21 | VSS_12 | VSS_13 | VSS_14 | VSS_15 | VSS_16 | VSS_17 |
| L | P11_D3P | P11_D3N | VSS_25 | Ref_filt_2 | VSS_26 | VDD_25 | VDD_26 | VSS_27 | VSS_28 | VSS_29 | VSS_30 | VSS_31 | VSS_32 |
| M | P11_D2P | P11_D2N | VDD_AH_31 | VDD_AH_32 | VDD_AH_33 | VDD_29 | VDD_30 | VSS_41 | VSS_42 | VSS_43 | VSS_44 | VSS_45 | VSS_46 |
| N | P11_D1P | P11_D1N | VSS_53 | VSS_54 | VSS_55 | VDD_33 | VDD_34 | VSS_56 | VSS_57 | VSS_58 | VSS_59 | VSS_60 | VSS_61 |
| P | P11_D0P | P11_D0N | VSS_71 | Reserved_24 | VDD_IO_7 | VDD_37 | VDD_38 | VSS_72 | VSS_73 | VSS_74 | VSS_75 | VSS_76 | VSS_77 |
| R | GPIO_31 | GPIO_30 | GPIO_29 | GPIO_28 | VDD_IO_8 | VDD_41 | VDD_42 | VSS_86 | VSS_87 | VSS_88 | VSS_89 | VSS_90 | VSS_91 |
| T | GPIO_27 | GPIO_26 | GPIO_25 | GPIO_24 | VDD_IO_9 | VDD_45 | VDD_46 | VSS_98 | VSS_99 | VSS_100 | VSS_101 | VSS_102 | VSS_103 |
| U | GPIO_23 | GPIO_22 | GPIO_21 | GPIO_20 | VDD_IO_10 | VSS_110 | VSS_111 | VSS_112 | VSS_113 | VSS_114 | VSS_115 | VSS_116 | VSS_117 |
| V | GPIO_19 | GPIO_18 | GPIO_17 | GPIO_16 | VDD_IO_11 | VDD_49 | VDD_50 | VDD_51 | VDD_52 | VDD_53 | VDD_54 | VDD_55 | VDD_56 |
| W | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | VDD_IO_12 | VDD_65 | VDD_66 | VDD_67 | VDD_68 | VDD_69 | VDD_70 | VDD_71 | VDD_72 |
| Y | GPIO_11 | GPIO_10 | GPIO_9 | GPIO_8 | VDD_IO_13 | Reserved_146 | Reserved_141 | RefClk_P | SerDes_E3_TxP | Reserved_134 | Reserved_129 | VSS_126 | SerDes_E2_TxP |
| AA | GPIO_7 | GPIO_6 | GPIO_5 | GPIO_4 | VDD_IO_14 | Reserved_147 | Reserved_140 | RefClk_N | SerDes_E3_TxN | Reserved_135 | Reserved_128 | VSS_145 | SerDes_E2_TxN |
| AB | GPIO_3 | GPIO_2 | GPIO_1 | GPIO_0 | VDD_IO_15 | VSS_129 | VSS_130 | VSS_131 | VSS_132 | VSS_133 | VSS_134 | VSS_135 | VSS_136 |
| AC | SI_DO | SI_nEn | VSS_148 | VDD_IO_16 | VDD_IO_17 | VDD_A_1 | VDD_A_2 | VDD_A_3 | VDD_A_4 | VDD_A_5 | VDD_A_6 | VDD_A_7 | VDD_A_8 |
| AD | SI_Clk | SI_DI | Reserved_18 | VDD_IO_18 | VSS_149 | VDD_VS_1 | VDD_VS_2 | VDD_VS_3 | VDD_VS_4 | VDD_VS_5 | VDD_VS_6 | VDD_VS_7 | VDD_VS_8 |
| AE | VSS_151 | Reserved_17 | VDD_IO_19 | VSS_163 | VSS_152 | Reserved_144 | Reserved_143 | Reserved_22 | SerDes_E3_RxP | Reserved_132 | Reserved_131 | VSS_153 | SerDes_E2_RxP |
| AF | | VDD_IO_20 | MDIO | MDC | VSS_158 | Reserved_145 | Reserved_142 | Reserved_23 | SerDes_E3_RxN | Reserved_133 | Reserved_130 | VSS_159 | SerDes_E2_RxN |

***Figure 111*** • **Pin Diagram for VSC7427-02, Top Right**

| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P5_D0P | P5_D1P | P5_D2P | P5_D3P | P4_D0P | P4_D1P | P4_D2P | P4_D3P | P3_D0P | P3_D1P | P3_D2P | P3_D3P | | A |
| P5_D0N | P5_D1N | P5_D2N | P5_D3N | P4_D0N | P4_D1N | P4_D2N | P4_D3N | P3_D0N | P3_D1N | P3_D2N | P3_D3N | VSS_2 | B |
| RefClk_Sel2 | Reserved_8 | Reserved_7 | Reserved_6 | Reserved_5 | Reserved_201 | Reserved_202 | Reserved_203 | THERMDC_VSS | THERMDA | Reserved_204 | P2_D0N | P2_D0P | C |
| Reserved_12 | Reserved_212 | VDD_AH_5 | JTAG_CLK | JTAG_DI | JTAG_DO | JTAG_TMS | JTAG_TRST | Reserved_213 | Reserved_214 | Reserved_215 | P2_D1N | P2_D1P | D |
| Ref_filt_1 | VDD_AH_12 | VDD_AH_13 | VDD_AL_3 | VDD_AL_4 | VDD_AH_14 | VDD_IO_3 | VDD_IO_4 | VDD_AH_15 | VDD_AH_16 | Reserved_217 | P2_D2N | P2_D2P | E |
| Reserved_220 | VDD_AH_22 | VDD_AH_23 | VDD_AL_7 | VDD_AL_8 | VDD_AH_24 | VDD_AH_6 | VDD_IO_6 | VDD_AH_25 | VDD_AH_26 | Reserved_221 | P2_D3N | P2_D3P | F |
| Reserved_223 | VDD_6 | VDD_7 | VDD_AL_11 | VDD_AL_12 | VDD_8 | VDD_9 | VDD_10 | VSS_5 | Reserved_10 | VSS_6 | P1_D0N | P1_D0P | G |
| Reserved_225 | VDD_18 | VDD_19 | VDD_20 | VDD_21 | VDD_22 | VDD_23 | VDD_24 | VSS_9 | Reserved_11 | VSS_10 | P1_D1N | P1_D1P | H |
| Reserved_232 | Reserved_233 | Reserved_234 | Reserved_235 | Reserved_236 | Reserved_237 | VDD_AL_16 | VDD_AL_17 | VDD_AL_18 | VDD_AH_29 | VDD_AH_30 | P1_D2N | P1_D2P | J |
| VSS_18 | VSS_19 | VSS_20 | VSS_21 | VSS_22 | VSS_23 | VDD_AL_22 | VDD_AL_23 | VDD_AL_24 | Ref_rext_0 | VSS_24 | P1_D3N | P1_D3P | K |
| VSS_33 | VSS_34 | VSS_35 | VSS_36 | VSS_37 | VSS_38 | VDD_27 | VDD_28 | VSS_39 | Ref_filt_0 | VSS_40 | P0_D0N | P0_D0P | L |
| VSS_47 | VSS_48 | VSS_49 | VSS_50 | VSS_51 | VSS_52 | VDD_31 | VDD_32 | VDD_AH_34 | VDD_AH_35 | VDD_AH_36 | P0_D1N | P0_D1P | M |
| VSS_62 | VSS_63 | VSS_64 | VSS_65 | VSS_66 | VSS_67 | VDD_35 | VDD_36 | VSS_68 | VSS_69 | VSS_70 | P0_D2N | P0_D2P | N |
| VSS_78 | VSS_79 | VSS_80 | VSS_81 | VSS_82 | VSS_83 | VDD_39 | VDD_40 | VDD_IODDR_1 | VSS_84 | VSS_85 | P0_D3N | P0_D3P | P |
| VSS_92 | VSS_93 | VSS_94 | VSS_95 | VSS_96 | VSS_97 | VDD_43 | VDD_44 | VDD_IODDR_2 | Reserved_20 | Reserved_19 | DDR_Rext | DDR_Vref | R |
| VSS_104 | VSS_105 | VSS_106 | VSS_107 | VSS_108 | VSS_109 | VDD_47 | VDD_48 | VDD_IODDR_3 | Reserved_21 | DDR_A13 | DDR_A12 | DDR_A11 | T |
| VSS_118 | VSS_119 | VSS_120 | VSS_121 | VSS_122 | VSS_123 | VSS_124 | VSS_125 | VDD_IODDR_4 | DDR_A7 | DDR_A9 | DDR_A6 | DDR_A8 | U |
| VDD_57 | VDD_58 | VDD_59 | VDD_60 | VDD_61 | VDD_62 | VDD_63 | VDD_64 | VDD_IODDR_5 | DDR_A3 | DDR_A5 | DDR_A2 | DDR_A4 | V |
| VDD_73 | VDD_74 | VDD_75 | VDD_76 | VDD_77 | VDD_78 | VDD_79 | VDD_80 | VDD_IODDR_6 | DDR_A10 | DDR_A1 | DDR_nCAS | DDR_A0 | W |
| Reserved_121 | Reserved_118 | VSS_127 | SerDes_E1_TxP | Reserved_110 | SerDes0_TxP | VSS_128 | SerDes_E0_TxP | VDD_IODDR_7 | DDR_BA0 | DDR_BA1 | DDR_ODT | DDR_nRAS | Y |
| Reserved_120 | Reserved_119 | VSS_146 | SerDes_E1_TxN | Reserved_111 | SerDes0_TxN | VSS_147 | SerDes_E0_TxN | VDD_IODDR_8 | DDR_nWE | DDR_BA2 | DDR_CK | DDR_CKn | AA |
| VSS_137 | VSS_138 | VSS_139 | VSS_140 | VSS_141 | VSS_142 | VSS_143 | VSS_144 | VDD_IODDR_9 | DDR_DQ3 | DDR_CKE | DDR_DQ2 | DDR_DQ5 | AB |
| VDD_A_9 | VDD_A_10 | VDD_A_11 | VDD_A_12 | VDD_A_13 | VDD_A_14 | VDD_A_15 | VDD_A_16 | VDD_IODDR_10 | DDR_DQ1 | DDR_DQ4 | DDR_DQ7 | DDR_DQ0 | AC |
| VDD_VS_9 | VDD_VS_10 | VDD_VS_11 | VDD_VS_12 | VDD_VS_13 | VDD_VS_14 | VDD_VS_15 | VDD_VS_16 | VSS_150 | VDD_IODDR_11 | DDR_DQ6 | DDR_DQS | DDR_DQSn | AD |
| Reserved_123 | Reserved_116 | VSS_154 | SerDes_E1_RxP | Reserved_108 | SerDes0_RxP | VSS_155 | SerDes_E0_RxP | SerDes_Rext_0 | VSS_156 | VDD_IODDR_12 | DDR_DM | VSS_157 | AE |
| Reserved_122 | Reserved_117 | VSS_160 | SerDes_E1_RxN | Reserved_109 | SerDes0_RxN | VSS_161 | SerDes_E0_RxN | SerDes_Rext_1 | VSS_162 | VDD_IODDR_14 | VDD_IODDR_13 | | AF |

# 12.2 Pins by Function for VSC7427-02

This section contains the functional pin descriptions for the VSC7427-02 device. The following table lists the definitions for the pin type symbols.

***Table 897*** • **Pin Type Symbol Definitions**

| Symbol | Pin Type | Description |
|---|---|---|
| ABIAS | Analog bias | Analog bias pin. |
| DIFF | Differential | Differential signal pair. |
| I | Input | Input signal. |
| O | Output | Output signal. |
| I/O | Bidirectional | Bidirectional input or output signal. |
| A | Analog input | Analog input for sensing variable voltage levels. |
| PD | Pull-down | On-chip pull-down resistor to VSS. |
| PU | Pull-up | On-chip pull-up resistor to VDD_IO. |
| 3V | | 3.3 V-tolerant. |
| O | Output | Output signal. |

*Table 897 •* **Pin Type Symbol Definitions *(continued)***

| Symbol | Pin Type | Description |
|---|---|---|
| OZ | 3-state output | Output. |
| ST | Schmitt-trigger | Input has Schmitt-trigger circuitry. |
| TD | Termination differential | Internal differential termination. |

## 12.2.1 Analog Bias Signals

The following table lists the pins associated with the analog bias signals.

*Table 898 •* **Analog Bias Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_Rext_[1:0] | A | Analog bias calibration.<br>Connect an external 620 Ω ±1% resistor between SerDes_Rext_1 and SerDes_Rext_0. |
| Ref_filt_[2:0] | A | Reference filter. Connect a 1.0 µF external capacitor between each pin and ground. |
| Ref_rext_[2:0] | A | Reference external resistor. Connect a 2.0 kΩ (1%) resistor between each pin and ground. |

## 12.2.2 Clock Circuits

The following table lists the pins associated with the system clock interface.

*Table 899 •* **System Clock Interface Pins**

| Name | Type | Description |
|---|---|---|
| RefClk_Sel[2:0] | I, PD | Reference clock frequency selection.<br>0: Connect to pull-down or leave floating.<br>1: Connect to pull-up to $V_{DD_IO}$.<br>Coding:<br>000: 125 MHz (default).<br>001: 156.25 MHz.<br>010: Reserved.<br>011: Reserved.<br>100: 25 MHz.<br>101: Reserved.<br>110: Reserved.<br>111: Reserved. |
| RefClk_P<br>RefClk_N | I, Diff | Reference clock input.<br>The input can be either differential or single-ended. In differential mode, REFCLK_P is the true part of the differential signal, and REFCLK_N is the complement part of the differential signal.<br>In single-ended mode, REFCLK_P is used as single-ended LVTTL input, and the REFCLK_N should be pulled to $V_{DD_A}$.<br>Required applied frequency depends on RefClk_Sel[2:0] input state. See description for RefClk_Sel[2:0] pins. |

### 12.2.3 DDR2 SDRAM Interface

The following table lists the pins associated with the DDR2 SDRAM interface.

***Table 900 •*** **DDR2 SDRAM Pins**

| Name | Type | Description |
|---|---|---|
| DDR_CK<br>DDR_CKn | 0, Diff | SDRAM differential clock. Differential clock to external SDRAM.<br>DDR_CK is the true part of the differential signal. DDR_nCk is the complement part. |
| DDR_CKE | O | SDRAM clock enable.<br>0: Disables clock in external SDRAM.<br>1: Enables clock in external SDRAM. |
| DDR_nRAS<br>DDR_nCAS<br>DDR_nWE | O | SDRAM command outputs.<br>DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered. |
| DDR_DM | O | SDRAM data mask outputs.<br>DDR_DM and DDR_UDM are mask signals for data written to the SDRAM. DDR_LDM corresponds to data on DDR_DQ[7:0], and DDR_UDM corresponds to data on DDR_DQ[15:8]. |
| DDR_BA[2:0] | O | SDRAM bank address outputs.<br>DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied. |
| DDR_A[13:0] | O | SDRAM address outputs.<br>Provide row and column addresses to the SDRAM. |
| DDR_DQ[7:0] | I/O | SDRAM data bus. |
| DDR_DQS<br>DDR_DQSn | I/O, Diff | SDRAM differential data strobes.<br>Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. |
| DDR_ODT | O | Control signals for the attached DDR2 SDRAM devices on-die termination. |
| DDR_Rext | ABIAS | External DDR impedance calibration.<br>Connect the pin through an external 240 Ω ±1% resistor to ground. |
| DDR_Vref | ABIAS | Input reference voltage.<br>Provides the input switching reference voltage to the SSTL DDR signals. |

### 12.2.4 General-Purpose Inputs and Outputs

The following table lists the pins associated with general-purpose inputs and outputs. The GPIO pins have an alternate function signal, which is mapped out in the following table. The overlaid functions are

selected by software on a pin-by-pin basis. The parallel interface and MIIM slave interface are enabled depending on the VCORE_CFG settings and override the normal GPIO and alternate functions.

***Table 901 •*** **GPIO Pin Mapping**

| Name | Overlaid Function 1 | Overlaid Function 2 | Parallel Interface | MIIM Slave Interface | Type |
|---|---|---|---|---|---|
| GPIO_0 | SIO_CLK | | | | I/O, PU,ST, 3V |
| GPIO_1 | SIO_LD | | | | I/O, PU,ST, 3V |
| GPIO_2 | SIO_DO | | | | I/O, PU,ST, 3V |
| GPIO_3 | SIO_DI | | | | I/O, PU,ST, 3V |
| GPIO_4 | TACHO | | | | I/O, PU,ST, 3V |
| GPIO_5 | TWI_SCL | PHY0_LED1 | | | I/O, PU,ST, 3V |
| GPIO_6 | TWI_SDA | PHY1_LED1 | | | I/O, PU,ST, 3V |
| GPIO_7 | | PHY2_LED1 | | | I/O, PU,ST, 3V |
| GPIO_8 | EXT_IRQ0 | PHY3_LED1 | | | I/O, PU,ST, 3V |
| GPIO_9 | EXT_IRQ1 | PHY4_LED1 | | | I/O, PU,ST, 3V |
| GPIO_10 | SFP14_SD | PHY5_LED1 | | | I/O, PU,ST, 3V |
| GPIO_11 | SFP15_SD | PHY6_LED1 | | | I/O, PU,ST, 3V |
| GPIO_12 | SFP17_SD | PHY7_LED1 | | | I/O, PU,ST, 3V |
| GPIO_13 | SFP18_SD | PHY8_LED1 | PI_nCS | | I/O, PU,ST, 3V |
| GPIO_14 | SI_nEN1 | PHY9_LED1 | PI_nWR | SLV_ADDR | I/O, PU,ST, 3V |
| GPIO_15 | SI_nEn2 | PHY10_LED1 | PI_nOE | SLV_MDC | I/O, PU,ST, 3V |
| GPIO_16 | SI_nEn3 | PHY11_LED1 | PI_nDone | SLV_MDIO | I/O, PU,ST, 3V |
| GPIO_17 | SFP10_SD | PHY0_LED0 | PI_A0 | | I/O, PU,ST, 3V |
| GPIO_18 | SFP11_SD | PHY2_LED0 | PI_A1 | | I/O, PU,ST, 3V |
| GPIO_19 | SFP12_SD | PHY2_LED0 | PI_A2 | | I/O, PU,ST, 3V |
| GPIO_20 | SFP13_SD | PHY3_LED0 | PI_A3 | | I/O, PU,ST, 3V |
| GPIO_21 | SFP16_SD | PHY4_LED0 | PI_D0 | | I/O, PU,ST, 3V |
| GPIO_22 | SFP19_SD | PHY5_LED0 | PI_D1 | | I/O, PU,ST, 3V |
| GPIO_23 | SFP24_SD | PHY6_LED0 | PI_D2 | | I/O, PU,ST, 3V |
| GPIO_24 | SFP25_SD | PHY7_LED0 | PI_D3 | | I/O, PU,ST, 3V |
| GPIO_25 | SFP20_SD | PHY8_LED0 | PI_D4 | | I/O, PU,ST, 3V |
| GPIO_26 | SFP21_SD | PHY9_LED0 | PI_D5 | | I/O, PU,ST, 3V |
| GPIO_27 | SFP22_SD | PHY10_LED0 | PI_D6 | | I/O, PU,ST, 3V |
| GPIO_28 | SFP23_SD | PHY11_LED0 | PI_D7 | | I/O, PU,ST, 3V |
| GPIO_29 | PWM | | | | I/O, PU,ST, 3V |
| GPIO_30 | UART_TX | | | | I/O, PU,ST, 3V |
| GPIO_31 | UART_RX | | | | I/O, PU,ST, 3V |

## 12.2.5 JTAG Interface

The following table lists the pins associated with the JTAG interface. The JTAG interface can be connected to the boundary scan TAP controller or the internal VCore-III TAP controller for software debug as described for the VCore_ICE_nEn signal.

The JTAG signals are not 5 V tolerant.

***Table 902 •*** **JTAG Interface Pins**

| Name | Type | Description |
|---|---|---|
| JTAG_nTRST | I, PU, ST, 3V | JTAG test reset, active low. For normal device operation, JTAG_nTRST should be pulled low. |
| JTAG_CLK | I, PU, ST, 3V | JTAG clock. |
| JTAG_TDI | I, PU, ST, 3V | JTAG test data in. |
| JTAG_TDO | OZ, 3V | JTAG test data out. |
| JTAG_TMS | I, PU, ST, 3V | JTAG test mode select. |

## 12.2.6 MII Management Interface

The following table lists the pins associated with the MII Management interface.

***Table 903 •*** **MII Management Interface Pins**

| Name | Type | Description |
|---|---|---|
| MDIO | I/O, 3V | Management data input/output.<br>MDIO is a bidirectional signal between a PHY and the device, used to transfer control and status information. Control information is driven by the device synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by the device. |
| MDC | O, 3V | Management data clock.<br>MDC is sourced by the station management entity (the device) to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal. |

## 12.2.7 Miscellaneous Signals

The following table lists the pins associated with a particular interface or facility on the device.

***Table 904 •*** **Miscellaneous Pins**

| Name | Type | Description |
|---|---|---|
| nReset | I, PD, ST, 3V | Global device reset, active low. |
| COMA_MODE | I/O, PU, ST, 3V | When this pin is asserted high, all PHYs are held in a powered-down state.<br>When this pin is deasserted low, all PHYs are powered up and resume normal operation. Additionally, this signal is used to synchronize the operation of multiple devices on the same printed circuit board to provide visual synchronization for LEDs driven from the separate devices. |

*Table 904 •* **Miscellaneous Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| VCORE_CFG[2:0] | I, PD, ST, 3V | Configuration signals for controlling the VCore-III CPU functions. |
| VCore_ICE_nEn | I, PU, 3V | VCore ICE nEn.<br>0: Enables the VCore-III JTAG debug interface over the JTAG interface pins.<br>1: Enables normal IO-JTAG over the JTAG interface. |
| THERMDA | A | Thermal diode anode (p-junction). |
| THERMDC_VSS | A | Thermal diode cathode (n-junction). Connected on-die to $V_{SS}$. |
| EXT_IRQ[1:0](1) | I/O PD, 3V | This pin interrupts inputs or outputs to the internal VCore-III CPU system or to an external processor. Signal polarity is programmable. |
| Reserved_1<br>Reserved_7 | I, PD, ST, 3V | Tie to $V_{DD_IO}$. |
| Reserved_4<br>Reserved_[5:6]<br>Reserved_8 | I, PD, ST, 3V | Tie to $V_{SS}$. |
| Reserved_[10:15]<br>Reserved_[17:24]<br>Reserved_[108:111]<br>Reserved_[116:123]<br>Reserved_[128:135]<br>Reserved_[140:147]<br>Reserved_[201:209]<br>Reserved_[211:221]<br>Reserved_[223]<br>Reserved_[225]<br>Reserved_[232:237]<br>Reserved_[240:248] | I, PD, ST, 3V | Leave floating. |

1. Available as an alternate function on the GPIO_8 and GPIO_9 pins.

## 12.2.8 Parallel Interface

The parallel interface (PI) can operate in a Master mode or a Slave mode according to the VCORE_CFG[1:0] signal settings. In Master mode, the internal VCore-III CPU system controls the PI and can access external peripherals over it. In Slave mode, the PI can be used by an external CPU to access internal device resources.

The PI master and slave mode signals are alternate function signals on GPIO pins. For more information about the GPIO mapping, see Table 901, page 710.

*Table 905 •* **Parallel Interface VCore-III Master Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | OZ, 3V | External address bus.<br>Used for addressing external memory space. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | External data bus. PI_Data0 is LSB. |

*Table 905 •* **Parallel Interface VCore-III Master Mode Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| PI_nCS | OZ, 3V | Programmable active low chip selects.<br>PI_nCS is used as default for booting from external memory (typically Flash). |
| PI_nDone | I, 3V | Acknowledges an operation. Used for external device-paced access operation.<br>Signal polarity is programmable. |
| PI_nOE | OZ, 3V | Active low signal that signals external device to drive data bus during read access. |
| PI_nWR | OZ, 3V | Active low signal that signals external access direction. Read (1) or write (0). |

The following pins are associated with the parallel CPU interface slave mode.

*Table 906 •* **Parallel CPU Interface Slave Mode Pins**

| Name | Type | Description |
|---|---|---|
| PI_Addr[3:0] | I, 3V | Internal device register address bus. Controlled by external CPU. PI_Addr0 is LSB. |
| PI_Data[7:0] | I/O, 3V | Data bus.<br>PI_Data[0] is LSB. |
| PI_nCS | I, 3V | Device chip select. |
| PI_nDone | O, 3V | Acknowledges an operation. Signal polarity is programmable. |
| PI_nOE | I, 3V | Signals device to drive data bus during read operations. |
| PI_nWR | I, 3V | Signals access direction. Read (1) or write (0). |

## 12.2.9 Power Supplies and Ground

The following table lists the power supply and ground pins.

*Table 907 •* **Power Supply and Ground Pins**

| Name | Type | Description |
|---|---|---|
| VDD | Power | 1.0 V power supply voltage for core |
| VDD_A | Power | 1.0 V power supply voltage for analog circuits |
| VDD_AL | Power | 1.0 V power supply voltage for analog circuits for twisted pair interface |
| VDD_AH | Power | 2.5 V power supply voltage for analog driver in twisted pair interface |
| VDD_IO | Power | 2.5 V power supply for parallel CPU interface, MII Management interface, and miscellaneous I/Os |
| VDD_IODDR | Power | 1.8 V power supply for DDR interface |
| VDD_VS | Power | 1.0 V or 1.2 V power supply for SerDes and Enhanced SerDes interfaces |
| VSS | Ground | Ground reference |

## 12.2.10 Serial CPU Interface

The serial CPU interface (SI) can be used as a serial slave, master, or boot interface.

As a slave interface, it allows external CPUs to access internal registers. As a master interface, it allows the device to access external devices using a programmable protocol. The serial CPU interface also allows the internal VCore-III CPU system to boot from an attached serial memory device when the VCORE_CFG signals are set appropriately.

The following table lists the pins associated with the serial CPU interface (SI).

***Table 908 •*** **Serial CPU Interface Pins**

| Name | Type | Description |
|---|---|---|
| SI_Clk | I/O, 3V | Slave mode: Input receiving serial interface clock from external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven with clock to external serial memory device. |
| SI_DI | I, 3V | Slave mode: Input receiving serial interface data from external master.<br>Master mode: Input directly read by software from register bit.<br>Boot mode: Input boot data from external serial memory device. |
| SI_DO | OZ, 3V | Slave mode: Output transmitting serial interface data to external master.<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: No function. |
| SI_nEn<br>SI_nEn[3:1](1) | I/O, 3V | Slave mode: Input used to enable SI slave interface.<br>0: Enabled<br>1: Disabled<br>Master mode: Output controlled directly by software through register bit.<br>Boot mode: Output driven while booting from EEPROM or serial flash to internal VCore-III CPU system. Released when booting is completed. |

1. Available as an alternate function on the GPIO_16, GPIO_15, and GPIO_14 pins. For more information about GPIO pin mapping, see Table 901, page 710.

## 12.2.11 SerDes Interface

The following pins are associated with the SerDes (SGMII) interface.

***Table 909 •*** **SerDes Interface Pins**

| Name | Type | Description |
|---|---|---|
| SerDes0_RxP, N | I, Diff, TD | Differential SerDes data inputs. |
| SerDes0_TxP, N | O, Diff | Differential SerDes data outputs. |

## 12.2.12 Enhanced SerDes Interface

The following pins are associated with the Enhanced SerDes interface.

***Table 910 •*** **Enhanced SerDes Interface Pins**

| Name | Type | Description |
|---|---|---|
| SerDes_E[0:3]_RxP, N | I, Diff, TD | Differential SGMII data inputs. |

*Table 910 •* **Enhanced SerDes Interface Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| SerDes_E[0:3]_TxP, N | O, Diff | Differential SGMII data outputs. |

## 12.2.13 Twisted Pair Interface

The following pins are associated with the twisted pair interface. The PHYn_LED[1:0] LED control signals associated with the twisted pair interfaces are alternate functions on the GPIOs. For more information about the GPIO pin mapping, see Table 901, page 710.

*Table 911 •* **Twisted Pair Interface Pins**

| Name | Type | Description |
|---|---|---|
| P0_D0P<br>P1_D0P<br>P2_D0P<br>P3_D0P<br>P4_D0P<br>P5_D0P<br>P6_D0P<br>P7_D0P<br>P8_D0P<br>P9_D0P<br>P10_D0P<br>P11_D0P | $A_{DIFF}$ | Tx/Rx channel A positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1. |
| P0_D0N<br>P1_D0N<br>P2_D0N<br>P3_D0N<br>P4_D0N<br>P5_D0N<br>P6_D0N<br>P7_D0N<br>P8_D0N<br>P9_D0N<br>P10_D0N<br>P11_D0N | $A_{DIFF}$ | Tx/Rx channel A negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2. |
| P0_D1P<br>P1_D1P<br>P2_D1P<br>P3_D1P<br>P4_D1P<br>P5_D1P<br>P6_D1P<br>P7_D1P<br>P8_D1P<br>P9_D1P<br>P10_D1P<br>P11_D1P | $A_{DIFF}$ | Tx/Rx channel B positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3. |

*Table 911 •* **Twisted Pair Interface Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| P0_D1N<br>P1_D1N<br>P2_D1N<br>P3_D1N<br>P4_D1N<br>P5_D1N<br>P6_D1N<br>P7_D1N<br>P8_D1N<br>P9_D1N<br>P10_D1N<br>P11_D1N | $A_{DIFF}$ | Tx/Rx channel B negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6. |
| P0_D2P<br>P1_D2P<br>P2_D2P<br>P3_D2P<br>P4_D2P<br>P5_D2P<br>P6_D2P<br>P7_D2P<br>P8_D2P<br>P9_D2P<br>P10_D2P<br>P11_D2P | $A_{DIFF}$ | Tx/Rx channel C positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes). |
| P0_D2N<br>P1_D2N<br>P2_D2N<br>P3_D2N<br>P4_D2N<br>P5_D2N<br>P6_D2N<br>P7_D2N<br>P8_D2N<br>P9_D2N<br>P10_D2N<br>P11_D2N | $A_{DIFF}$ | Tx/Rx channel C negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes). |
| P0_D3P<br>P1_D3P<br>P2_D3P<br>P3_D3P<br>P4_D3P<br>P5_D3P<br>P6_D3P<br>P7_D3P<br>P8_D3P<br>P9_D3P<br>P10_D3P<br>P11_D3P | $A_{DIFF}$ | Tx/Rx channel D positive signal.<br>Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes). |

*Table 911 •* **Twisted Pair Interface Pins *(continued)***

| Name | Type | Description |
|---|---|---|
| P0_D3N<br>P1_D3N<br>P2_D3N<br>P3_D3N<br>P4_D3N<br>P5_D3N<br>P6_D3N<br>P7_D3N<br>P8_D3N<br>P9_D3N<br>P10_D3N<br>P11_D3N | $A_{DIFF}$ | Tx/Rx channel D negative signal.<br>Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes). |

## 12.3 Pins by Number for VSC7427-02

This section provides a numeric list of the VSC7427-02 pins.

| | |
|---|---|
| A2 | P8_D0P |
| A3 | P8_D1P |
| A4 | P8_D2P |
| A5 | P8_D3P |
| A6 | P7_D0P |
| A7 | P7_D1P |
| A8 | P7_D2P |
| A9 | P7_D3P |
| A10 | P6_D0P |
| A11 | P6_D1P |
| A12 | P6_D2P |
| A13 | P6_D3P |
| A14 | P5_D0P |
| A15 | P5_D1P |
| A16 | P5_D2P |
| A17 | P5_D3P |
| A18 | P4_D0P |
| A19 | P4_D1P |
| A20 | P4_D2P |
| A21 | P4_D3P |
| A22 | P3_D0P |
| A23 | P3_D1P |
| A24 | P3_D2P |
| A25 | P3_D3P |
| AA1 | GPIO_7 |
| AA2 | GPIO_6 |
| AA3 | GPIO_5 |
| AA4 | GPIO_4 |
| AA5 | VDD_IO_14 |
| AA6 | Reserved_147 |
| AA7 | Reserved_140 |
| AA8 | RefClk_N |
| AA9 | SerDes_E3_TxN |
| AA10 | Reserved_135 |
| AA11 | Reserved_128 |
| AA12 | VSS_145 |
| AA13 | SerDes_E2_TxN |
| AA14 | Reserved_120 |
| AA15 | Reserved_119 |
| AA16 | VSS_146 |
| AA17 | SerDes_E1_TxN |
| AA18 | Reserved_111 |
| AA19 | SerDes0_TxN |
| AA20 | VSS_147 |
| AA21 | SerDes_E0_TxN |
| AA22 | VDD_IODDR_8 |
| AA23 | DDR_nWE |
| AA24 | DDR_BA2 |
| AA25 | DDR_CK |
| AA26 | DDR_CKn |
| AB1 | GPIO_3 |
| AB2 | GPIO_2 |
| AB3 | GPIO_1 |
| AB4 | GPIO_0 |
| AB5 | VDD_IO_15 |
| AB6 | VSS_129 |
| AB7 | VSS_130 |
| AB8 | VSS_131 |
| AB9 | VSS_132 |
| AB10 | VSS_133 |
| AB11 | VSS_134 |
| AB12 | VSS_135 |
| AB13 | VSS_136 |
| AB14 | VSS_137 |
| AB15 | VSS_138 |
| AB16 | VSS_139 |
| AB17 | VSS_140 |
| AB18 | VSS_141 |
| AB19 | VSS_142 |
| AB20 | VSS_143 |
| AB21 | VSS_144 |
| AB22 | VDD_IODDR_9 |
| AB23 | DDR_DQ3 |
| AB24 | DDR_CKE |
| AB25 | DDR_DQ2 |
| AB26 | DDR_DQ5 |
| AC1 | SI_DO |
| AC2 | SI_nEn |
| AC3 | VSS_148 |
| AC4 | VDD_IO_16 |
| AC5 | VDD_IO_17 |
| AC6 | VDD_A_1 |
| AC7 | VDD_A_2 |
| AC8 | VDD_A_3 |
| AC9 | VDD_A_4 |
| AC10 | VDD_A_5 |
| AC11 | VDD_A_6 |
| AC12 | VDD_A_7 |
| AC13 | VDD_A_8 |
| AC14 | VDD_A_9 |
| AC15 | VDD_A_10 |
| AC16 | VDD_A_11 |
| AC17 | VDD_A_12 |
| AC18 | VDD_A_13 |
| AC19 | VDD_A_14 |
| AC20 | VDD_A_15 |
| AC21 | VDD_A_16 |
| AC22 | VDD_IODDR_10 |
| AC23 | DDR_DQ1 |
| AC24 | DDR_DQ4 |
| AC25 | DDR_DQ7 |
| AC26 | DDR_DQ0 |
| AD1 | SI_Clk |
| AD2 | SI_DI |
| AD3 | Reserved_18 |
| AD4 | VDD_IO_18 |
| AD5 | VSS_149 |
| AD6 | VDD_VS_1 |
| AD7 | VDD_VS_2 |
| AD8 | VDD_VS_3 |

Pins by number *(continued)*

| | |
|---|---|
| AD9 | VDD_VS_4 |
| AD10 | VDD_VS_5 |
| AD11 | VDD_VS_6 |
| AD12 | VDD_VS_7 |
| AD13 | VDD_VS_8 |
| AD14 | VDD_VS_9 |
| AD15 | VDD_VS_10 |
| AD16 | VDD_VS_11 |
| AD17 | VDD_VS_12 |
| AD18 | VDD_VS_13 |
| AD19 | VDD_VS_14 |
| AD20 | VDD_VS_15 |
| AD21 | VDD_VS_16 |
| AD22 | VSS_150 |
| AD23 | VDD_IODDR_11 |
| AD24 | DDR_DQ6 |
| AD25 | DDR_DQS |
| AD26 | DDR_DQSn |
| AE1 | VSS_151 |
| AE2 | Reserved_17 |
| AE3 | VDD_IO_19 |
| AE4 | VSS_163 |
| AE5 | VSS_152 |
| AE6 | Reserved_144 |
| AE7 | Reserved_143 |
| AE8 | Reserved_22 |
| AE9 | SerDes_E3_RxP |
| AE10 | Reserved_132 |
| AE11 | Reserved_131 |
| AE12 | VSS_153 |
| AE13 | SerDes_E2_RxP |
| AE14 | Reserved_123 |
| AE15 | Reserved_116 |
| AE16 | VSS_154 |
| AE17 | SerDes_E1_RxP |
| AE18 | Reserved_108 |
| AE19 | SerDes0_RxP |
| AE20 | VSS_155 |
| AE21 | SerDes_E0_RxP |
| AE22 | SerDes_Rext_0 |
| AE23 | VSS_156 |
| AE24 | VDD_IODDR_12 |
| AE25 | DDR_DM |
| AE26 | VSS_157 |
| AF2 | VDD_IO_20 |
| AF3 | MDIO |
| AF4 | MDC |
| AF5 | VSS_158 |
| AF6 | Reserved_145 |
| AF7 | Reserved_142 |
| AF8 | Reserved_23 |
| AF9 | SerDes_E3_RxN |
| AF10 | Reserved_133 |
| AF11 | Reserved_130 |
| AF12 | VSS_159 |
| AF13 | SerDes_E2_RxN |
| AF14 | Reserved_122 |
| AF15 | Reserved_117 |
| AF16 | VSS_160 |
| AF17 | SerDes_E1_RxN |
| AF18 | Reserved_109 |
| AF19 | SerDes0_RxN |
| AF20 | VSS_161 |
| AF21 | SerDes_E0_RxN |
| AF22 | SerDes_Rext_1 |
| AF23 | VSS_162 |
| AF24 | VDD_IODDR_14 |
| AF25 | VDD_IODDR_13 |
| B1 | VSS_1 |
| B2 | P8_D0N |
| B3 | P8_D1N |
| B4 | P8_D2N |
| B5 | P8_D3N |
| B6 | P7_D0N |
| B7 | P7_D1N |
| B8 | P7_D2N |
| B9 | P7_D3N |
| B10 | P6_D0N |
| B11 | P6_D1N |
| B12 | P6_D2N |
| B13 | P6_D3N |
| B14 | P5_D0N |
| B15 | P5_D1N |
| B16 | P5_D2N |
| B17 | P5_D3N |
| B18 | P4_D0N |
| B19 | P4_D1N |
| B20 | P4_D2N |
| B21 | P4_D3N |
| B22 | P3_D0N |
| B23 | P3_D1N |
| B24 | P3_D2N |
| B25 | P3_D3N |
| B26 | VSS_2 |
| C1 | P9_D3P |
| C2 | P9_D3N |
| C3 | COMA_MODE |
| C4 | nRESET |
| C5 | VDD_IO_21 |
| C6 | Reserved_1 |
| C7 | VCORE_CFG0 |
| C8 | VCORE_CFG1 |
| C9 | VCORE_CFG2 |
| C10 | VCore_ICE_nEn |
| C11 | Reserved_4 |
| C12 | RefClk_Sel0 |
| C13 | RefClk_Sel1 |
| C14 | RefClk_Sel2 |
| C15 | Reserved_8 |
| C16 | Reserved_7 |
| C17 | Reserved_6 |
| C18 | Reserved_5 |
| C19 | Reserved_201 |
| C20 | Reserved_202 |
| C21 | Reserved_203 |
| C22 | THERMDC_VSS |
| C23 | THERMDA |

Pins by number *(continued)*

| | |
|---|---|
| C24 | Reserved_204 |
| C25 | P2_D0N |
| C26 | P2_D0P |
| D1 | P9_D2P |
| D2 | P9_D2N |
| D3 | Reserved_205 |
| D4 | VDD_AH_1 |
| D5 | VDD_AH_2 |
| D6 | Reserved_206 |
| D7 | Reserved_207 |
| D8 | Reserved_208 |
| D9 | Reserved_209 |
| D10 | Reserved_248 |
| D11 | VDD_AH_4 |
| D12 | Reserved_211 |
| D13 | Reserved_13 |
| D14 | Reserved_12 |
| D15 | Reserved_212 |
| D16 | VDD_AH_5 |
| D17 | JTAG_CLK |
| D18 | JTAG_DI |
| D19 | JTAG_DO |
| D20 | JTAG_TMS |
| D21 | JTAG_TRST |
| D22 | Reserved_213 |
| D23 | Reserved_214 |
| D24 | Reserved_215 |
| D25 | P2_D1N |
| D26 | P2_D1P |
| E1 | P9_D1P |
| E2 | P9_D1N |
| E3 | Reserved_216 |
| E4 | VDD_AH_7 |
| E5 | VDD_AH_8 |
| E6 | VDD_IO_1 |
| E7 | VDD_IO_2 |
| E8 | VDD_AH_9 |
| E9 | VDD_AL_1 |
| E10 | VDD_AL_2 |
| E11 | VDD_AH_10 |
| E12 | VDD_AH_11 |
| E13 | Ref_rext_1 |
| E14 | Ref_filt_1 |
| E15 | VDD_AH_12 |
| E16 | VDD_AH_13 |
| E17 | VDD_AL_3 |
| E18 | VDD_AL_4 |
| E19 | VDD_AH_14 |
| E20 | VDD_IO_3 |
| E21 | VDD_IO_4 |
| E22 | VDD_AH_15 |
| E23 | VDD_AH_16 |
| E24 | Reserved_217 |
| E25 | P2_D2N |
| E26 | P2_D2P |
| F1 | P9_D0P |
| F2 | P9_D0N |
| F3 | Reserved_218 |
| F4 | VDD_AH_17 |
| F5 | VDD_AH_18 |
| F6 | VDD_IO_5 |
| F7 | VDD_AH_3 |
| F8 | VDD_AH_19 |
| F9 | VDD_AL_5 |
| F10 | VDD_AL_6 |
| F11 | VDD_AH_20 |
| F12 | VDD_AH_21 |
| F13 | Reserved_219 |
| F14 | Reserved_220 |
| F15 | VDD_AH_22 |
| F16 | VDD_AH_23 |
| F17 | VDD_AL_7 |
| F18 | VDD_AL_8 |
| F19 | VDD_AH_24 |
| F20 | VDD_AH_6 |
| F21 | VDD_IO_6 |
| F22 | VDD_AH_25 |
| F23 | VDD_AH_26 |
| F24 | Reserved_221 |
| F25 | P2_D3N |
| F26 | P2_D3P |
| G1 | P10_D3P |
| G2 | P10_D3N |
| G3 | VSS_3 |
| G4 | Reserved_15 |
| G5 | VSS_4 |
| G6 | VDD_1 |
| G7 | VDD_2 |
| G8 | VDD_3 |
| G9 | VDD_AL_9 |
| G10 | VDD_AL_10 |
| G11 | VDD_4 |
| G12 | VDD_5 |
| G13 | Reserved_247 |
| G14 | Reserved_223 |
| G15 | VDD_6 |
| G16 | VDD_7 |
| G17 | VDD_AL_11 |
| G18 | VDD_AL_12 |
| G19 | VDD_8 |
| G20 | VDD_9 |
| G21 | VDD_10 |
| G22 | VSS_5 |
| G23 | Reserved_10 |
| G24 | VSS_6 |
| G25 | P1_D0N |
| G26 | P1_D0P |
| H1 | P10_D2P |
| H2 | P10_D2N |
| H3 | VSS_7 |
| H4 | Reserved_14 |
| H5 | VSS_8 |
| H6 | VDD_11 |
| H7 | VDD_12 |
| H8 | VDD_13 |
| H9 | VDD_14 |
| H10 | VDD_15 |

Pins by number *(continued)*

| | |
|---|---|
| H11 | VDD_16 |
| H12 | VDD_17 |
| H13 | Reserved_246 |
| H14 | Reserved_225 |
| H15 | VDD_18 |
| H16 | VDD_19 |
| H17 | VDD_20 |
| H18 | VDD_21 |
| H19 | VDD_22 |
| H20 | VDD_23 |
| H21 | VDD_24 |
| H22 | VSS_9 |
| H23 | Reserved_11 |
| H24 | VSS_10 |
| H25 | P1_D1N |
| H26 | P1_D1P |
| J1 | P10_D1P |
| J2 | P10_D1N |
| J3 | VDD_AH_27 |
| J4 | VDD_AH_28 |
| J5 | VDD_AL_13 |
| J6 | VDD_AL_14 |
| J7 | VDD_AL_15 |
| J8 | Reserved_240 |
| J9 | Reserved_241 |
| J10 | Reserved_242 |
| J11 | Reserved_243 |
| J12 | Reserved_244 |
| J13 | Reserved_245 |
| J14 | Reserved_232 |
| J15 | Reserved_233 |
| J16 | Reserved_234 |
| J17 | Reserved_235 |
| J18 | Reserved_236 |
| J19 | Reserved_237 |
| J20 | VDD_AL_16 |
| J21 | VDD_AL_17 |
| J22 | VDD_AL_18 |
| J23 | VDD_AH_29 |
| J24 | VDD_AH_30 |
| J25 | P1_D2N |
| J26 | P1_D2P |
| K1 | P10_D0P |
| K2 | P10_D0N |
| K3 | VSS_11 |
| K4 | Ref_rext_2 |
| K5 | VDD_AL_19 |
| K6 | VDD_AL_20 |
| K7 | VDD_AL_21 |
| K8 | VSS_12 |
| K9 | VSS_13 |
| K10 | VSS_14 |
| K11 | VSS_15 |
| K12 | VSS_16 |
| K13 | VSS_17 |
| K14 | VSS_18 |
| K15 | VSS_19 |
| K16 | VSS_20 |
| K17 | VSS_21 |
| K18 | VSS_22 |
| K19 | VSS_23 |
| K20 | VDD_AL_22 |
| K21 | VDD_AL_23 |
| K22 | VDD_AL_24 |
| K23 | Ref_rext_0 |
| K24 | VSS_24 |
| K25 | P1_D3N |
| K26 | P1_D3P |
| L1 | P11_D3P |
| L2 | P11_D3N |
| L3 | VSS_25 |
| L4 | Ref_filt_2 |
| L5 | VSS_26 |
| L6 | VDD_25 |
| L7 | VDD_26 |
| L8 | VSS_27 |
| L9 | VSS_28 |
| L10 | VSS_29 |
| L11 | VSS_30 |
| L12 | VSS_31 |
| L13 | VSS_32 |
| L14 | VSS_33 |
| L15 | VSS_34 |
| L16 | VSS_35 |
| L17 | VSS_36 |
| L18 | VSS_37 |
| L19 | VSS_38 |
| L20 | VDD_27 |
| L21 | VDD_28 |
| L22 | VSS_39 |
| L23 | Ref_filt_0 |
| L24 | VSS_40 |
| L25 | P0_D0N |
| L26 | P0_D0P |
| M1 | P11_D2P |
| M2 | P11_D2N |
| M3 | VDD_AH_31 |
| M4 | VDD_AH_32 |
| M5 | VDD_AH_33 |
| M6 | VDD_29 |
| M7 | VDD_30 |
| M8 | VSS_41 |
| M9 | VSS_42 |
| M10 | VSS_43 |
| M11 | VSS_44 |
| M12 | VSS_45 |
| M13 | VSS_46 |
| M14 | VSS_47 |
| M15 | VSS_48 |
| M16 | VSS_49 |
| M17 | VSS_50 |
| M18 | VSS_51 |
| M19 | VSS_52 |
| M20 | VDD_31 |
| M21 | VDD_32 |
| M22 | VDD_AH_34 |
| M23 | VDD_AH_35 |

Pins by number *(continued)*

| | |
|---|---|
| M24 | VDD_AH_36 |
| M25 | P0_D1N |
| M26 | P0_D1P |
| N1 | P11_D1P |
| N2 | P11_D1N |
| N3 | VSS_53 |
| N4 | VSS_54 |
| N5 | VSS_55 |
| N6 | VDD_33 |
| N7 | VDD_34 |
| N8 | VSS_56 |
| N9 | VSS_57 |
| N10 | VSS_58 |
| N11 | VSS_59 |
| N12 | VSS_60 |
| N13 | VSS_61 |
| N14 | VSS_62 |
| N15 | VSS_63 |
| N16 | VSS_64 |
| N17 | VSS_65 |
| N18 | VSS_66 |
| N19 | VSS_67 |
| N20 | VDD_35 |
| N21 | VDD_36 |
| N22 | VSS_68 |
| N23 | VSS_69 |
| N24 | VSS_70 |
| N25 | P0_D2N |
| N26 | P0_D2P |
| P1 | P11_D0P |
| P2 | P11_D0N |
| P3 | VSS_71 |
| P4 | Reserved_24 |
| P5 | VDD_IO_7 |
| P6 | VDD_37 |
| P7 | VDD_38 |
| P8 | VSS_72 |
| P9 | VSS_73 |
| P10 | VSS_74 |

| | |
|---|---|
| P11 | VSS_75 |
| P12 | VSS_76 |
| P13 | VSS_77 |
| P14 | VSS_78 |
| P15 | VSS_79 |
| P16 | VSS_80 |
| P17 | VSS_81 |
| P18 | VSS_82 |
| P19 | VSS_83 |
| P20 | VDD_39 |
| P21 | VDD_40 |
| P22 | VDD_IODDR_1 |
| P23 | VSS_84 |
| P24 | VSS_85 |
| P25 | P0_D3N |
| P26 | P0_D3P |
| R1 | GPIO_31 |
| R2 | GPIO_30 |
| R3 | GPIO_29 |
| R4 | GPIO_28 |
| R5 | VDD_IO_8 |
| R6 | VDD_41 |
| R7 | VDD_42 |
| R8 | VSS_86 |
| R9 | VSS_87 |
| R10 | VSS_88 |
| R11 | VSS_89 |
| R12 | VSS_90 |
| R13 | VSS_91 |
| R14 | VSS_92 |
| R15 | VSS_93 |
| R16 | VSS_94 |
| R17 | VSS_95 |
| R18 | VSS_96 |
| R19 | VSS_97 |
| R20 | VDD_43 |
| R21 | VDD_44 |
| R22 | VDD_IODDR_2 |
| R23 | Reserved_20 |

| | |
|---|---|
| R24 | Reserved_19 |
| R25 | DDR_Rext |
| R26 | DDR_Vref |
| T1 | GPIO_27 |
| T2 | GPIO_26 |
| T3 | GPIO_25 |
| T4 | GPIO_24 |
| T5 | VDD_IO_9 |
| T6 | VDD_45 |
| T7 | VDD_46 |
| T8 | VSS_98 |
| T9 | VSS_99 |
| T10 | VSS_100 |
| T11 | VSS_101 |
| T12 | VSS_102 |
| T13 | VSS_103 |
| T14 | VSS_104 |
| T15 | VSS_105 |
| T16 | VSS_106 |
| T17 | VSS_107 |
| T18 | VSS_108 |
| T19 | VSS_109 |
| T20 | VDD_47 |
| T21 | VDD_48 |
| T22 | VDD_IODDR_3 |
| T23 | Reserved_21 |
| T24 | DDR_A13 |
| T25 | DDR_A12 |
| T26 | DDR_A11 |
| U1 | GPIO_23 |
| U2 | GPIO_22 |
| U3 | GPIO_21 |
| U4 | GPIO_20 |
| U5 | VDD_IO_10 |
| U6 | VSS_110 |
| U7 | VSS_111 |
| U8 | VSS_112 |
| U9 | VSS_113 |
| U10 | VSS_114 |

Pins by number *(continued)*

| | |
|---|---|
| U11 | VSS_115 |
| U12 | VSS_116 |
| U13 | VSS_117 |
| U14 | VSS_118 |
| U15 | VSS_119 |
| U16 | VSS_120 |
| U17 | VSS_121 |
| U18 | VSS_122 |
| U19 | VSS_123 |
| U20 | VSS_124 |
| U21 | VSS_125 |
| U22 | VDD_IODDR_4 |
| U23 | DDR_A7 |
| U24 | DDR_A9 |
| U25 | DDR_A6 |
| U26 | DDR_A8 |
| V1 | GPIO_19 |
| V2 | GPIO_18 |
| V3 | GPIO_17 |
| V4 | GPIO_16 |
| V5 | VDD_IO_11 |
| V6 | VDD_49 |
| V7 | VDD_50 |
| V8 | VDD_51 |
| V9 | VDD_52 |
| V10 | VDD_53 |
| V11 | VDD_54 |
| V12 | VDD_55 |
| V13 | VDD_56 |
| V14 | VDD_57 |
| V15 | VDD_58 |
| V16 | VDD_59 |
| V17 | VDD_60 |
| V18 | VDD_61 |
| V19 | VDD_62 |
| V20 | VDD_63 |
| V21 | VDD_64 |
| V22 | VDD_IODDR_5 |
| V23 | DDR_A3 |
| V24 | DDR_A5 |
| V25 | DDR_A2 |
| V26 | DDR_A4 |
| W1 | GPIO_15 |
| W2 | GPIO_14 |
| W3 | GPIO_13 |
| W4 | GPIO_12 |
| W5 | VDD_IO_12 |
| W6 | VDD_65 |
| W7 | VDD_66 |
| W8 | VDD_67 |
| W9 | VDD_68 |
| W10 | VDD_69 |
| W11 | VDD_70 |
| W12 | VDD_71 |
| W13 | VDD_72 |
| W14 | VDD_73 |
| W15 | VDD_74 |
| W16 | VDD_75 |
| W17 | VDD_76 |
| W18 | VDD_77 |
| W19 | VDD_78 |
| W20 | VDD_79 |
| W21 | VDD_80 |
| W22 | VDD_IODDR_6 |
| W23 | DDR_A10 |
| W24 | DDR_A1 |
| W25 | DDR_nCAS |
| W26 | DDR_A0 |
| Y1 | GPIO_11 |
| Y2 | GPIO_10 |
| Y3 | GPIO_9 |
| Y4 | GPIO_8 |
| Y5 | VDD_IO_13 |
| Y6 | Reserved_146 |
| Y7 | Reserved_141 |
| Y8 | RefClk_P |
| Y9 | SerDes_E3_TxP |
| Y10 | Reserved_134 |
| Y11 | Reserved_129 |
| Y12 | VSS_126 |
| Y13 | SerDes_E2_TxP |
| Y14 | Reserved_121 |
| Y15 | Reserved_118 |
| Y16 | VSS_127 |
| Y17 | SerDes_E1_TxP |
| Y18 | Reserved_110 |
| Y19 | SerDes0_TxP |
| Y20 | VSS_128 |
| Y21 | SerDes_E0_TxP |
| Y22 | VDD_IODDR_7 |
| Y23 | DDR_BA0 |
| Y24 | DDR_BA1 |
| Y25 | DDR_ODT |
| Y26 | DDR_nRAS |

## 12.4 Pins by Name for VSC7427-02

This section provides a alphabetical list of the VSC7427-02 pins.

| | |
|---|---|
| COMA_MODE | C3 |
| DDR_A0 | W26 |
| DDR_A1 | W24 |
| DDR_A2 | V25 |
| DDR_A3 | V23 |
| DDR_A4 | V26 |
| DDR_A5 | V24 |
| DDR_A6 | U25 |
| DDR_A7 | U23 |
| DDR_A8 | U26 |
| DDR_A9 | U24 |
| DDR_A10 | W23 |
| DDR_A11 | T26 |
| DDR_A12 | T25 |
| DDR_A13 | T24 |
| DDR_BA0 | Y23 |
| DDR_BA1 | Y24 |
| DDR_BA2 | AA24 |
| DDR_CK | AA25 |
| DDR_CKE | AB24 |
| DDR_CKn | AA26 |
| DDR_DM | AE25 |
| DDR_DQ0 | AC26 |
| DDR_DQ1 | AC23 |
| DDR_DQ2 | AB25 |
| DDR_DQ3 | AB23 |
| DDR_DQ4 | AC24 |
| DDR_DQ5 | AB26 |
| DDR_DQ6 | AD24 |
| DDR_DQ7 | AC25 |
| DDR_DQS | AD25 |
| DDR_DQSn | AD26 |
| DDR_nCAS | W25 |
| DDR_nRAS | Y26 |
| DDR_nWE | AA23 |
| DDR_ODT | Y25 |
| DDR_Rext | R25 |
| DDR_Vref | R26 |
| GPIO_0 | AB4 |
| GPIO_1 | AB3 |
| GPIO_2 | AB2 |
| GPIO_3 | AB1 |
| GPIO_4 | AA4 |
| GPIO_5 | AA3 |
| GPIO_6 | AA2 |
| GPIO_7 | AA1 |
| GPIO_8 | Y4 |
| GPIO_9 | Y3 |
| GPIO_10 | Y2 |
| GPIO_11 | Y1 |
| GPIO_12 | W4 |
| GPIO_13 | W3 |
| GPIO_14 | W2 |
| GPIO_15 | W1 |
| GPIO_16 | V4 |
| GPIO_17 | V3 |
| GPIO_18 | V2 |
| GPIO_19 | V1 |
| GPIO_20 | U4 |
| GPIO_21 | U3 |
| GPIO_22 | U2 |
| GPIO_23 | U1 |
| GPIO_24 | T4 |
| GPIO_25 | T3 |
| GPIO_26 | T2 |
| GPIO_27 | T1 |
| GPIO_28 | R4 |
| GPIO_29 | R3 |
| GPIO_30 | R2 |
| GPIO_31 | R1 |
| JTAG_CLK | D17 |
| JTAG_DI | D18 |
| JTAG_DO | D19 |
| JTAG_TMS | D20 |
| JTAG_TRST | D21 |
| MDC | AF4 |
| MDIO | AF3 |
| nRESET | C4 |
| P0_D0N | L25 |
| P0_D0P | L26 |
| P0_D1N | M25 |
| P0_D1P | M26 |
| P0_D2N | N25 |
| P0_D2P | N26 |
| P0_D3N | P25 |
| P0_D3P | P26 |
| P1_D0N | G25 |
| P1_D0P | G26 |
| P1_D1N | H25 |
| P1_D1P | H26 |
| P1_D2N | J25 |
| P1_D2P | J26 |
| P1_D3N | K25 |
| P1_D3P | K26 |
| P2_D0N | C25 |
| P2_D0P | C26 |
| P2_D1N | D25 |
| P2_D1P | D26 |
| P2_D2N | E25 |
| P2_D2P | E26 |
| P2_D3N | F25 |
| P2_D3P | F26 |
| P3_D0N | B22 |
| P3_D0P | A22 |
| P3_D1N | B23 |
| P3_D1P | A23 |
| P3_D2N | B24 |
| P3_D2P | A24 |
| P3_D3N | B25 |
| P3_D3P | A25 |

Pins by name *(continued)*

| | |
|---|---|
| P4_D0N | B18 |
| P4_D0P | A18 |
| P4_D1N | B19 |
| P4_D1P | A19 |
| P4_D2N | B20 |
| P4_D2P | A20 |
| P4_D3N | B21 |
| P4_D3P | A21 |
| P5_D0N | B14 |
| P5_D0P | A14 |
| P5_D1N | B15 |
| P5_D1P | A15 |
| P5_D2N | B16 |
| P5_D2P | A16 |
| P5_D3N | B17 |
| P5_D3P | A17 |
| P6_D0N | B10 |
| P6_D0P | A10 |
| P6_D1N | B11 |
| P6_D1P | A11 |
| P6_D2N | B12 |
| P6_D2P | A12 |
| P6_D3N | B13 |
| P6_D3P | A13 |
| P7_D0N | B6 |
| P7_D0P | A6 |
| P7_D1N | B7 |
| P7_D1P | A7 |
| P7_D2N | B8 |
| P7_D2P | A8 |
| P7_D3N | B9 |
| P7_D3P | A9 |
| P8_D0N | B2 |
| P8_D0P | A2 |
| P8_D1N | B3 |
| P8_D1P | A3 |
| P8_D2N | B4 |
| P8_D2P | A4 |
| P8_D3N | B5 |
| P8_D3P | A5 |
| P9_D0N | F2 |
| P9_D0P | F1 |
| P9_D1N | E2 |
| P9_D1P | E1 |
| P9_D2N | D2 |
| P9_D2P | D1 |
| P9_D3N | C2 |
| P9_D3P | C1 |
| P10_D0N | K2 |
| P10_D0P | K1 |
| P10_D1N | J2 |
| P10_D1P | J1 |
| P10_D2N | H2 |
| P10_D2P | H1 |
| P10_D3N | G2 |
| P10_D3P | G1 |
| P11_D0N | P2 |
| P11_D0P | P1 |
| P11_D1N | N2 |
| P11_D1P | N1 |
| P11_D2N | M2 |
| P11_D2P | M1 |
| P11_D3N | L2 |
| P11_D3P | L1 |
| Ref_filt_0 | L23 |
| Ref_filt_1 | E14 |
| Ref_filt_2 | L4 |
| Ref_rext_0 | K23 |
| Ref_rext_1 | E13 |
| Ref_rext_2 | K4 |
| RefClk_N | AA8 |
| RefClk_P | Y8 |
| RefClk_Sel0 | C12 |
| RefClk_Sel1 | C13 |
| RefClk_Sel2 | C14 |
| Reserved_1 | C6 |
| Reserved_4 | C11 |
| Reserved_5 | C18 |
| Reserved_6 | C17 |
| Reserved_7 | C16 |
| Reserved_8 | C15 |
| Reserved_10 | G23 |
| Reserved_11 | H23 |
| Reserved_12 | D14 |
| Reserved_13 | D13 |
| Reserved_14 | H4 |
| Reserved_15 | G4 |
| Reserved_17 | AE2 |
| Reserved_18 | AD3 |
| Reserved_19 | R24 |
| Reserved_20 | R23 |
| Reserved_21 | T23 |
| Reserved_22 | AE8 |
| Reserved_23 | AF8 |
| Reserved_24 | P4 |
| Reserved_108 | AE18 |
| Reserved_109 | AF18 |
| Reserved_110 | Y18 |
| Reserved_111 | AA18 |
| Reserved_116 | AE15 |
| Reserved_117 | AF15 |
| Reserved_118 | Y15 |
| Reserved_119 | AA15 |
| Reserved_120 | AA14 |
| Reserved_121 | Y14 |
| Reserved_122 | AF14 |
| Reserved_123 | AE14 |
| Reserved_128 | AA11 |
| Reserved_129 | Y11 |
| Reserved_130 | AF11 |
| Reserved_131 | AE11 |
| Reserved_132 | AE10 |
| Reserved_133 | AF10 |
| Reserved_134 | Y10 |
| Reserved_135 | AA10 |
| Reserved_140 | AA7 |
| Reserved_141 | Y7 |

Pins by name *(continued)*

| | |
|---|---|
| Reserved_142 | AF7 |
| Reserved_143 | AE7 |
| Reserved_144 | AE6 |
| Reserved_145 | AF6 |
| Reserved_146 | Y6 |
| Reserved_147 | AA6 |
| Reserved_201 | C19 |
| Reserved_202 | C20 |
| Reserved_203 | C21 |
| Reserved_204 | C24 |
| Reserved_205 | D3 |
| Reserved_206 | D6 |
| Reserved_207 | D7 |
| Reserved_208 | D8 |
| Reserved_209 | D9 |
| Reserved_211 | D12 |
| Reserved_212 | D15 |
| Reserved_213 | D22 |
| Reserved_214 | D23 |
| Reserved_215 | D24 |
| Reserved_216 | E3 |
| Reserved_217 | E24 |
| Reserved_218 | F3 |
| Reserved_219 | F13 |
| Reserved_220 | F14 |
| Reserved_221 | F24 |
| Reserved_223 | G14 |
| Reserved_225 | H14 |
| Reserved_232 | J14 |
| Reserved_233 | J15 |
| Reserved_234 | J16 |
| Reserved_235 | J17 |
| Reserved_236 | J18 |
| Reserved_237 | J19 |
| Reserved_240 | J8 |
| Reserved_241 | J9 |
| Reserved_242 | J10 |
| Reserved_243 | J11 |
| Reserved_244 | J12 |
| Reserved_245 | J13 |
| Reserved_246 | H13 |
| Reserved_247 | G13 |
| Reserved_248 | D10 |
| SerDes_E0_RxN | AF21 |
| SerDes_E0_RxP | AE21 |
| SerDes_E0_TxN | AA21 |
| SerDes_E0_TxP | Y21 |
| SerDes_E1_RxN | AF17 |
| SerDes_E1_RxP | AE17 |
| SerDes_E1_TxN | AA17 |
| SerDes_E1_TxP | Y17 |
| SerDes_E2_RxN | AF13 |
| SerDes_E2_RxP | AE13 |
| SerDes_E2_TxN | AA13 |
| SerDes_E2_TxP | Y13 |
| SerDes_E3_RxN | AF9 |
| SerDes_E3_RxP | AE9 |
| SerDes_E3_TxN | AA9 |
| SerDes_E3_TxP | Y9 |
| SerDes_Rext_0 | AE22 |
| SerDes_Rext_1 | AF22 |
| SerDes0_RxN | AF19 |
| SerDes0_RxP | AE19 |
| SerDes0_TxN | AA19 |
| SerDes0_TxP | Y19 |
| SI_Clk | AD1 |
| SI_DI | AD2 |
| SI_DO | AC1 |
| SI_nEn | AC2 |
| THERMDA | C23 |
| THERMDC_VSS | C22 |
| VCORE_CFG0 | C7 |
| VCORE_CFG1 | C8 |
| VCORE_CFG2 | C9 |
| VCore_ICE_nEn | C10 |
| VDD_1 | G6 |
| VDD_2 | G7 |
| VDD_3 | G8 |
| VDD_4 | G11 |
| VDD_5 | G12 |
| VDD_6 | G15 |
| VDD_7 | G16 |
| VDD_8 | G19 |
| VDD_9 | G20 |
| VDD_10 | G21 |
| VDD_11 | H6 |
| VDD_12 | H7 |
| VDD_13 | H8 |
| VDD_14 | H9 |
| VDD_15 | H10 |
| VDD_16 | H11 |
| VDD_17 | H12 |
| VDD_18 | H15 |
| VDD_19 | H16 |
| VDD_20 | H17 |
| VDD_21 | H18 |
| VDD_22 | H19 |
| VDD_23 | H20 |
| VDD_24 | H21 |
| VDD_25 | L6 |
| VDD_26 | L7 |
| VDD_27 | L20 |
| VDD_28 | L21 |
| VDD_29 | M6 |
| VDD_30 | M7 |
| VDD_31 | M20 |
| VDD_32 | M21 |
| VDD_33 | N6 |
| VDD_34 | N7 |
| VDD_35 | N20 |
| VDD_36 | N21 |
| VDD_37 | P6 |
| VDD_38 | P7 |
| VDD_39 | P20 |
| VDD_40 | P21 |
| VDD_41 | R6 |
| VDD_42 | R7 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_43 | R20 |
| VDD_44 | R21 |
| VDD_45 | T6 |
| VDD_46 | T7 |
| VDD_47 | T20 |
| VDD_48 | T21 |
| VDD_49 | V6 |
| VDD_50 | V7 |
| VDD_51 | V8 |
| VDD_52 | V9 |
| VDD_53 | V10 |
| VDD_54 | V11 |
| VDD_55 | V12 |
| VDD_56 | V13 |
| VDD_57 | V14 |
| VDD_58 | V15 |
| VDD_59 | V16 |
| VDD_60 | V17 |
| VDD_61 | V18 |
| VDD_62 | V19 |
| VDD_63 | V20 |
| VDD_64 | V21 |
| VDD_65 | W6 |
| VDD_66 | W7 |
| VDD_67 | W8 |
| VDD_68 | W9 |
| VDD_69 | W10 |
| VDD_70 | W11 |
| VDD_71 | W12 |
| VDD_72 | W13 |
| VDD_73 | W14 |
| VDD_74 | W15 |
| VDD_75 | W16 |
| VDD_76 | W17 |
| VDD_77 | W18 |
| VDD_78 | W19 |
| VDD_79 | W20 |
| VDD_80 | W21 |
| VDD_A_1 | AC6 |
| VDD_A_2 | AC7 |
| VDD_A_3 | AC8 |
| VDD_A_4 | AC9 |
| VDD_A_5 | AC10 |
| VDD_A_6 | AC11 |
| VDD_A_7 | AC12 |
| VDD_A_8 | AC13 |
| VDD_A_9 | AC14 |
| VDD_A_10 | AC15 |
| VDD_A_11 | AC16 |
| VDD_A_12 | AC17 |
| VDD_A_13 | AC18 |
| VDD_A_14 | AC19 |
| VDD_A_15 | AC20 |
| VDD_A_16 | AC21 |
| VDD_AH_1 | D4 |
| VDD_AH_2 | D5 |
| VDD_AH_3 | F7 |
| VDD_AH_4 | D11 |
| VDD_AH_5 | D16 |
| VDD_AH_6 | F20 |
| VDD_AH_7 | E4 |
| VDD_AH_8 | E5 |
| VDD_AH_9 | E8 |
| VDD_AH_10 | E11 |
| VDD_AH_11 | E12 |
| VDD_AH_12 | E15 |
| VDD_AH_13 | E16 |
| VDD_AH_14 | E19 |
| VDD_AH_15 | E22 |
| VDD_AH_16 | E23 |
| VDD_AH_17 | F4 |
| VDD_AH_18 | F5 |
| VDD_AH_19 | F8 |
| VDD_AH_20 | F11 |
| VDD_AH_21 | F12 |
| VDD_AH_22 | F15 |
| VDD_AH_23 | F16 |
| VDD_AH_24 | F19 |
| VDD_AH_25 | F22 |
| VDD_AH_26 | F23 |
| VDD_AH_27 | J3 |
| VDD_AH_28 | J4 |
| VDD_AH_29 | J23 |
| VDD_AH_30 | J24 |
| VDD_AH_31 | M3 |
| VDD_AH_32 | M4 |
| VDD_AH_33 | M5 |
| VDD_AH_34 | M22 |
| VDD_AH_35 | M23 |
| VDD_AH_36 | M24 |
| VDD_AL_1 | E9 |
| VDD_AL_2 | E10 |
| VDD_AL_3 | E17 |
| VDD_AL_4 | E18 |
| VDD_AL_5 | F9 |
| VDD_AL_6 | F10 |
| VDD_AL_7 | F17 |
| VDD_AL_8 | F18 |
| VDD_AL_9 | G9 |
| VDD_AL_10 | G10 |
| VDD_AL_11 | G17 |
| VDD_AL_12 | G18 |
| VDD_AL_13 | J5 |
| VDD_AL_14 | J6 |
| VDD_AL_15 | J7 |
| VDD_AL_16 | J20 |
| VDD_AL_17 | J21 |
| VDD_AL_18 | J22 |
| VDD_AL_19 | K5 |
| VDD_AL_20 | K6 |
| VDD_AL_21 | K7 |
| VDD_AL_22 | K20 |
| VDD_AL_23 | K21 |
| VDD_AL_24 | K22 |
| VDD_IO_1 | E6 |
| VDD_IO_2 | E7 |
| VDD_IO_3 | E20 |

Pins by name *(continued)*

| | |
|---|---|
| VDD_IO_4 | E21 |
| VDD_IO_5 | F6 |
| VDD_IO_6 | F21 |
| VDD_IO_7 | P5 |
| VDD_IO_8 | R5 |
| VDD_IO_9 | T5 |
| VDD_IO_10 | U5 |
| VDD_IO_11 | V5 |
| VDD_IO_12 | W5 |
| VDD_IO_13 | Y5 |
| VDD_IO_14 | AA5 |
| VDD_IO_15 | AB5 |
| VDD_IO_16 | AC4 |
| VDD_IO_17 | AC5 |
| VDD_IO_18 | AD4 |
| VDD_IO_19 | AE3 |
| VDD_IO_20 | AF2 |
| VDD_IO_21 | C5 |
| VDD_IODDR_1 | P22 |
| VDD_IODDR_2 | R22 |
| VDD_IODDR_3 | T22 |
| VDD_IODDR_4 | U22 |
| VDD_IODDR_5 | V22 |
| VDD_IODDR_6 | W22 |
| VDD_IODDR_7 | Y22 |
| VDD_IODDR_8 | AA22 |
| VDD_IODDR_9 | AB22 |
| VDD_IODDR_10 | AC22 |
| VDD_IODDR_11 | AD23 |
| VDD_IODDR_12 | AE24 |
| VDD_IODDR_13 | AF25 |
| VDD_IODDR_14 | AF24 |
| VDD_VS_1 | AD6 |
| VDD_VS_2 | AD7 |
| VDD_VS_3 | AD8 |
| VDD_VS_4 | AD9 |
| VDD_VS_5 | AD10 |
| VDD_VS_6 | AD11 |
| VDD_VS_7 | AD12 |
| VDD_VS_8 | AD13 |
| VDD_VS_9 | AD14 |
| VDD_VS_10 | AD15 |
| VDD_VS_11 | AD16 |
| VDD_VS_12 | AD17 |
| VDD_VS_13 | AD18 |
| VDD_VS_14 | AD19 |
| VDD_VS_15 | AD20 |
| VDD_VS_16 | AD21 |
| VSS_1 | B1 |
| VSS_2 | B26 |
| VSS_3 | G3 |
| VSS_4 | G5 |
| VSS_5 | G22 |
| VSS_6 | G24 |
| VSS_7 | H3 |
| VSS_8 | H5 |
| VSS_9 | H22 |
| VSS_10 | H24 |
| VSS_11 | K3 |
| VSS_12 | K8 |
| VSS_13 | K9 |
| VSS_14 | K10 |
| VSS_15 | K11 |
| VSS_16 | K12 |
| VSS_17 | K13 |
| VSS_18 | K14 |
| VSS_19 | K15 |
| VSS_20 | K16 |
| VSS_21 | K17 |
| VSS_22 | K18 |
| VSS_23 | K19 |
| VSS_24 | K24 |
| VSS_25 | L3 |
| VSS_26 | L5 |
| VSS_27 | L8 |
| VSS_28 | L9 |
| VSS_29 | L10 |
| VSS_30 | L11 |
| VSS_31 | L12 |
| VSS_32 | L13 |
| VSS_33 | L14 |
| VSS_34 | L15 |
| VSS_35 | L16 |
| VSS_36 | L17 |
| VSS_37 | L18 |
| VSS_38 | L19 |
| VSS_39 | L22 |
| VSS_40 | L24 |
| VSS_41 | M8 |
| VSS_42 | M9 |
| VSS_43 | M10 |
| VSS_44 | M11 |
| VSS_45 | M12 |
| VSS_46 | M13 |
| VSS_47 | M14 |
| VSS_48 | M15 |
| VSS_49 | M16 |
| VSS_50 | M17 |
| VSS_51 | M18 |
| VSS_52 | M19 |
| VSS_53 | N3 |
| VSS_54 | N4 |
| VSS_55 | N5 |
| VSS_56 | N8 |
| VSS_57 | N9 |
| VSS_58 | N10 |
| VSS_59 | N11 |
| VSS_60 | N12 |
| VSS_61 | N13 |
| VSS_62 | N14 |
| VSS_63 | N15 |
| VSS_64 | N16 |
| VSS_65 | N17 |
| VSS_66 | N18 |
| VSS_67 | N19 |
| VSS_68 | N22 |
| VSS_69 | N23 |

Pins by name *(continued)*

| | |
|---|---|
| VSS_70 | N24 |
| VSS_71 | P3 |
| VSS_72 | P8 |
| VSS_73 | P9 |
| VSS_74 | P10 |
| VSS_75 | P11 |
| VSS_76 | P12 |
| VSS_77 | P13 |
| VSS_78 | P14 |
| VSS_79 | P15 |
| VSS_80 | P16 |
| VSS_81 | P17 |
| VSS_82 | P18 |
| VSS_83 | P19 |
| VSS_84 | P23 |
| VSS_85 | P24 |
| VSS_86 | R8 |
| VSS_87 | R9 |
| VSS_88 | R10 |
| VSS_89 | R11 |
| VSS_90 | R12 |
| VSS_91 | R13 |
| VSS_92 | R14 |
| VSS_93 | R15 |
| VSS_94 | R16 |
| VSS_95 | R17 |
| VSS_96 | R18 |
| VSS_97 | R19 |
| VSS_98 | T8 |
| VSS_99 | T9 |
| VSS_100 | T10 |
| VSS_101 | T11 |
| VSS_102 | T12 |
| VSS_103 | T13 |
| VSS_104 | T14 |
| VSS_105 | T15 |
| VSS_106 | T16 |
| VSS_107 | T17 |
| VSS_108 | T18 |
| VSS_109 | T19 |
| VSS_110 | U6 |
| VSS_111 | U7 |
| VSS_112 | U8 |
| VSS_113 | U9 |
| VSS_114 | U10 |
| VSS_115 | U11 |
| VSS_116 | U12 |
| VSS_117 | U13 |
| VSS_118 | U14 |
| VSS_119 | U15 |
| VSS_120 | U16 |
| VSS_121 | U17 |
| VSS_122 | U18 |
| VSS_123 | U19 |
| VSS_124 | U20 |
| VSS_125 | U21 |
| VSS_126 | Y12 |
| VSS_127 | Y16 |
| VSS_128 | Y20 |
| VSS_129 | AB6 |
| VSS_130 | AB7 |
| VSS_131 | AB8 |
| VSS_132 | AB9 |
| VSS_133 | AB10 |
| VSS_134 | AB11 |
| VSS_135 | AB12 |
| VSS_136 | AB13 |
| VSS_137 | AB14 |
| VSS_138 | AB15 |
| VSS_139 | AB16 |
| VSS_140 | AB17 |
| VSS_141 | AB18 |
| VSS_142 | AB19 |
| VSS_143 | AB20 |
| VSS_144 | AB21 |
| VSS_145 | AA12 |
| VSS_146 | AA16 |
| VSS_147 | AA20 |
| VSS_148 | AC3 |
| VSS_149 | AD5 |
| VSS_150 | AD22 |
| VSS_151 | AE1 |
| VSS_152 | AE5 |
| VSS_153 | AE12 |
| VSS_154 | AE16 |
| VSS_155 | AE20 |
| VSS_156 | AE23 |
| VSS_157 | AE26 |
| VSS_158 | AF5 |
| VSS_159 | AF12 |
| VSS_160 | AF16 |
| VSS_161 | AF20 |
| VSS_162 | AF23 |
| VSS_163 | AE4 |

# 13 Package Information

VSC7424XJG-02, VSC7425XJG-02, VSC7426XJG-02, and VSC7427XJG-02 are packaged in a lead(Pb)-free, 672-pin, thermally enhanced, plastic ball grid array (BGA) with a 27 mm × 27 mm body size, 1 mm pin pitch, and 2.36 mm maximum height.

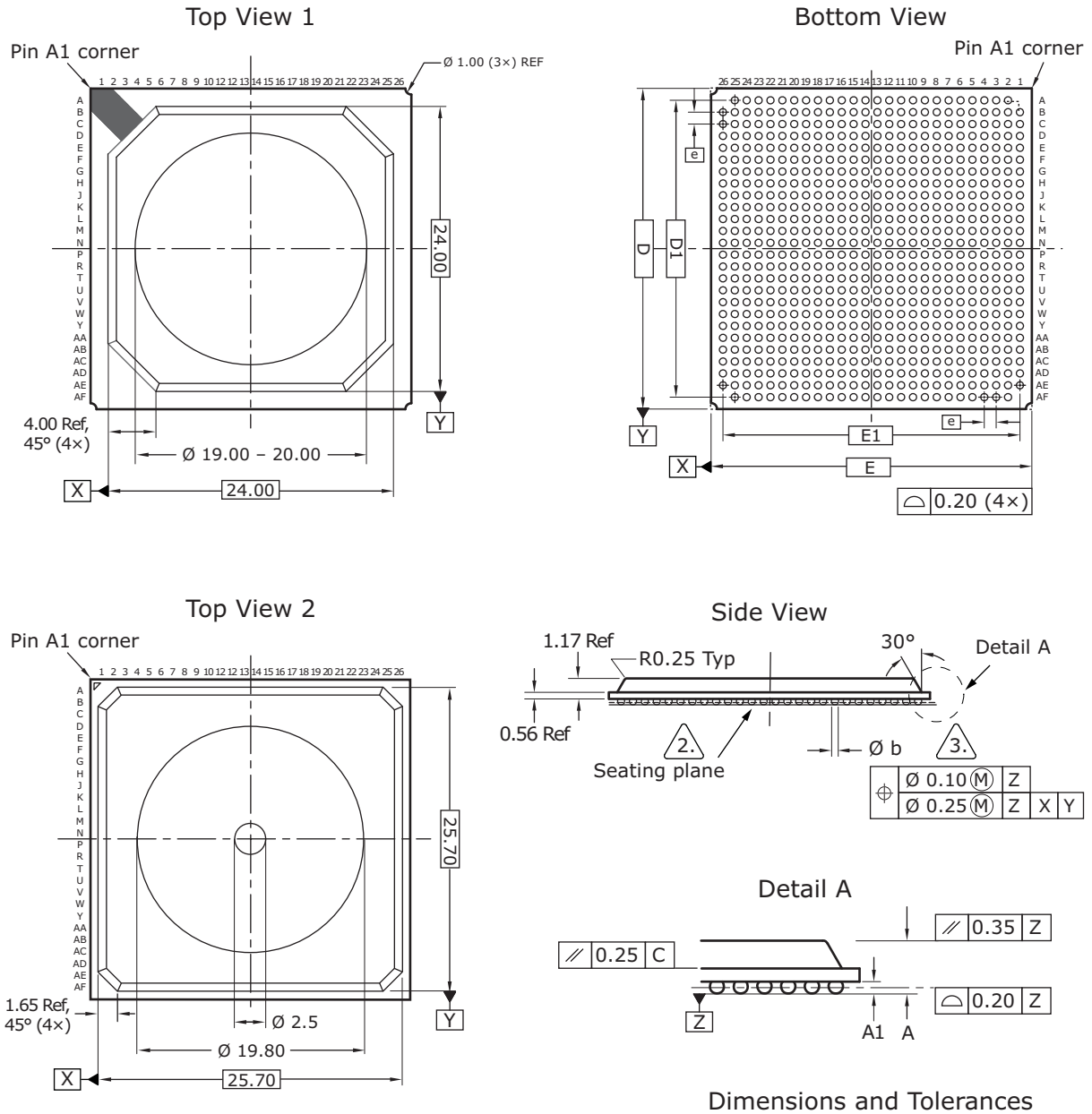
Lead-free products comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

This section provides the package drawing, thermal specifications, and moisture sensitivity rating for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

## 13.1 Package Drawing

The following illustration shows the package drawing for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices. The drawing contains the top view, bottom view, side view, detail view, dimensions, tolerances, and notes.

***Figure 112* • Package Drawing BGA**




Dimensions and Tolerances

| Reference | Minimum | Nominal | Maximum |
|---|---|---|---|
| A | 2.10 | 2.23 | 2.44 |
| A1 | 0.40 | 0.50 | 0.60 |
| D | | 27.00 | |
| E | | 27.00 | |
| D1 | | 25.00 | |
| E1 | | 25.00 | |
| e | | 1.00 | |
| 3. b | 0.50 | 0.60 | 0.70 |

Notes

1. All dimensions and tolerances are in millimeters (mm).
2. Primary datum Z and seating plane are defined by the spherical crowns of the solder balls.
3. Dimension is measured at the maximum solder ball diameter, parallel to primary datum C.
4. Radial true position is represented by typical values.
5. Top view 1 and top view 2 reflect one of two packages customers can expect to receive.

## 13.2 Thermal Specifications

Thermal specifications for these devices are based on the JEDEC JESD51 family of documents. These documents are available on the JEDEC Web site at www.jedec.org. The thermal specifications are

modeled using a four-layer test board with two signal layers, a power plane, and a ground plane (2s2p PCB). For more information about the thermal measurement method used for these devices, see the JESD51-1 standard.

***Table 912 •*** **Thermal Resistances BGA**

| Symbol | °C/W | Parameter |
|---|---|---|
| $\theta_{JCtop}$ | 3.27 | Die junction to package case top |
| $\theta_{JB}$ | 6.03 | Die junction to printed circuit board |
| $\theta_{JA}$ | 12.14 | Die junction to ambient |
| $\theta_{JMA}$ at 1 m/s | 9.42 | Die junction to moving air measured at an air speed of 1 m/s |
| $\theta_{JMA}$ at 2 m/s | 8 | Die junction to moving air measured at an air speed of 2 m/s |

To achieve results similar to the modeled thermal measurements, the guidelines for board design described in the JESD51 family of publications must be applied. For information about applications using BGA packages, see the following:

- JESD51-2A, *Integrated Circuits Thermal Test Method Environmental Conditions, Natural Convection (Still Air)*
- JESD51-6, *Integrated Circuit Thermal Test Method Environmental Conditions, Forced Convection (Moving Air)*
- JESD51-8, *Integrated Circuit Thermal Test Method Environmental Conditions, Junction-to-Board*
- JESD51-9, *Test Boards for Area Array Surface Mount Package Thermal Measurements*

## 13.3 Moisture Sensitivity

This device is rated moisture sensitivity level 4 as specified in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

# 14 Design Guidelines

This section provides information about design guidelines for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

## 14.1 Power Supplies

The following guidelines apply to designing power supplies for use with the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices:

- Make at least one unbroken ground plane (GND).
- Use the power and ground plane combination as an effective power supply bypass capacitor. The capacitance is proportional to the area of the two planes and inversely proportional to the separation between the planes. Typical values with a 0.25 mm (0.01 inch) separation are 100 pF/in2. This capacitance is more effective than a capacitor of equivalent value, because the planes have no inductance or Equivalent Series Resistance (ESR).
- Do not cut up the power or ground planes in an effort to steer current paths. This usually produces more noise, not less. Furthermore, place vias and clearances in a configuration that maintains the integrity of the plane. Groups of vias spaced close together often overlap clearances. This can form a large slot in the plane. As a result, return currents are forced around the slot, which increases the loop area and EMI emissions. Signals should never be placed on a ground plane, because the resulting slot forces return currents around the slot.
- Vias connecting power planes to the supply and ground balls must be at least 0.25 mm (0.010 inch) in diameter, preferably with no thermal relief and plated closed with copper or solder. Use separate (or even multiple) vias for each supply and ground ball.

## 14.2 Power Supply Decoupling

Each power supply voltage should have both bulk and high-frequency decoupling capacitors. Recommended capacitors are as follows:

- For bulk decoupling, use 10 µF high capacity and low ESR capacitors or equivalent, distributed across the board.
- For high-frequency decoupling, use 0.1 µF high frequency (for example, X7R) ceramic capacitors placed on the side of the PCB closest to the plane being decoupled, and as close as possible to the power ball. A larger value in the same housing unit produces even better results.
- Use surface-mounted components for lower lead inductance and pad capacitance. Smaller form factor components are best (that is, 0402 is better than 0603).

## 14.3 Reference Clock

The device reference clock can be a 25 MHz, 125 MHz, or 156.25 MHz clock signal. It can be either a differential reference clock or a single-ended clock. However, 25 MHz single-ended operation is not recommended when using QSGMII due to the jitter specification requirements of this interface. For more information, see Reference Clock, page 612.

### 14.3.1 Single-Ended RefClk Input

An external resistor network is required to use a single-ended reference clock. The network limits the amplitude and adjusts the center of the swing.

The following illustrations show configurations for a single-ended reference clock.

***Figure 113*** **• 2.5 V CMOS Single-Ended RefClk Input Resistor Network**

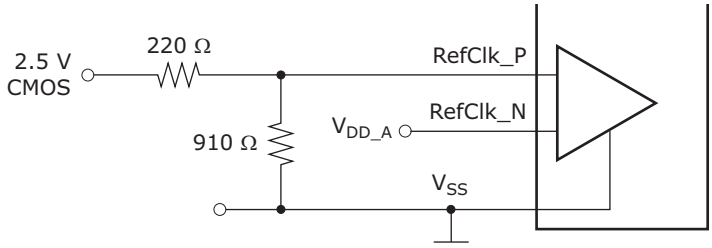
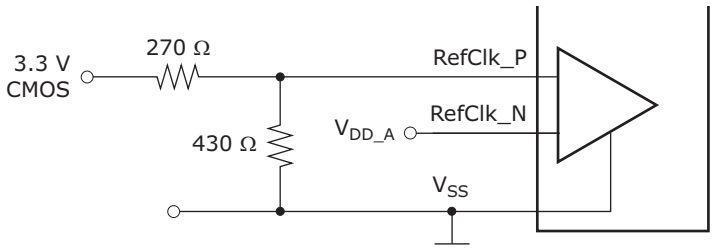



***Figure 114*** **• 3.3 V CMOS Single-Ended RefClk Input Resistor Network**




# 14.4 Interfaces

This section provides general recommendations for all interfaces and information related to the specific interfaces on the device.

## 14.4.1 General Recommendations

High-speed signals require excellent frequency and phase response up to the third harmonic. The best design would provide excellent frequency and phase response up to the seventh harmonic. The following recommendations can improve signal quality and minimize transmission distances:

Keep traces as short as possible. Initial component placement should be considered very carefully.

- The impedance of the traces must match the impedance of the termination resistors, connectors, and cable. This reduces reflections due to impedance mismatches.
- Differential impedance must be maintained in a 100 Ω differential application. Routing two 50 Ω traces is not adequate. The two traces must be separated by enough distance to maintain differential impedance. When routing differential pairs, keep the two trace lengths identical. Differences in trace lengths translate directly into signal skew. Note that the differential impedance may be affected when separations occur.
- Keep differential pair traces on the same layer of the PCB to minimize impedance discontinuities.
- Do not group all the passive components together. The pads of the components add capacitance to the traces. At the frequencies encountered, this can result in unwanted reductions in impedance. Use surface-mounted 0603 components to reduce this effect.
- Eliminate or reduce stub lengths.
- Reduce, if not eliminate, vias to minimize impedance discontinuities. Remember that vias and their clearance holes in the power/ground planes can cause impedance discontinuities in nearby signals. Keep vias away from other traces.
- Keep signal traces away from other signals that might capacitively couple noise into the signals. A good rule of thumb is to keep the traces apart by ten times the width of the trace.
- Do not route digital signals from other circuits across the area of the high-speed transmitter and receiver signals.
- Using grounded guard traces is typically not effective for improving signal quality. A ground plane is more effective. However, a common use of guard traces is to route them during the layout, but remove them prior to design completion. This has the benefit of enforcing keep-out areas around sensitive high-speed signals so that vias and other traces are not accidentally placed incorrectly.
- When signals in a differential pair are mismatched, the result is a common-mode current. In a well-designed system, common-mode currents should make up less than one percent of the total differential currents. Mode currents represent a primary source of EMI emissions. To reduce common-mode currents, route differential traces so that their lengths are the same. For example, a

5-mm (0.2-inch) length mismatch between differential signals having the rise and fall times of 200 ps results in the common-mode current being up to 18% of the differential current.

**Note** Care must be taken when choosing proper components (such as the termination resistors) in the designing of the layout of a printed circuit board, because of the high application frequency. The use of surface-mount components is highly recommended to minimize parasitic inductance and lead length of the termination resistor.

Matching the impedance of the PCB traces, connectors, and balanced interconnect media is also highly recommended. Impedance variations along the entire interconnect path must be minimized, because they degrade the signal path and may cause reflections of the signal.

## 14.4.2 SGMII Interface

The SGMII interface consists of a Tx and Rx differential pair operating at 1250 Mbps.

The SGMII signals can be routed on any PCB trace layer with the following constraints:

- The Tx output signals in a pair should have matched electrical lengths.
- The Rx input signals in a pair should have matched electrical lengths.
- SGMII Tx and Rx pairs must be routed as 100 Ω differential traces with ground plane as reference.
- Keep differential pair traces on the same layer of the PCB to minimize impedance discontinuities.
- AC-coupling of Tx and Rx may be needed, depending on the PHY. If AC-coupled, the inputs are self-biased.
- To reduce the crosstalk between pairs or other PCB lines, it is recommended that the spacing on each side of the pair be larger than four times the track width.

## 14.4.3 Parallel Interface

This section applies when the parallel interface is enabled.

The parallel interface (PI) consists of PI_Addr3:0], PI_Data[7:0], PI_nCS, PI_nDone, PI_nOE, and PI_nWR. Leave these signals floating if the parallel interface is not used.

When using the parallel interface, the timing parameter $t_{D(SLNH)}$ indicates when an issued command is sampled by the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02. For more information about the $t_{D(SLNH)}$ timing parameter, see Table 844, page 628.

To ensure that the PI_nDone signal is driven inactive properly, add a 4.7 kΩ pull-up resistor to this signal, when used.

## 14.4.4 Serial Interface

If the serial CPU interface is not used, all input signals can be left floating.

The SI bus consists of the SI_Clk clock signal, the SI_DO and SI_DI data signals, and the SI_nCS0 device select signal.

When routing the SI_Clk signal, be sure to create clean edges. If the SI bus is connected to more than one slave device, route it in a daisy-chain configuration with no stubs. Terminate the SI_Clk signal properly to avoid reflections and double clocking.

If it is not possible (or desirable) to route the bus in a daisy-chain configuration, the SI_Clk signal should be buffered and routed in a star topology from the buffers. Each buffered clock should be terminated at its source.

The SI tristates the SI_Clk and SI_DO signals prior to deasserting the SI_nCS0 signal. This makes it possible to implement CPOL/CPHA as 0/0 or 1/1, if the attached SI devices require it, using termination resistors. If the attached devices support both types of CPOL/CPHA, SI_Clk and SI_DO must still have pull resistors to one of the I/O supply rails to prevent spurious clocks being seen when the signals are tristated.

## 14.4.5 Enhanced SerDes Interface

The Enhanced SerDes interface can be used for fiber connections, backplanes, or direct coupler cable connections, such as the CX4 cable.

The Enhanced SerDes interface can operate in several modes. The physical signal bit rate is between 125 Mbps to 6.25 Mbps.

The inputs are self-biased and have internal AC-coupling. In some modes, the interface requires external AC-coupling, because of the input DC voltage limitation. If external AC-coupling capacitors are required, it is recommended to use small form factor components, such as 0603. The small form factor minimizes impedance mismatch by the AC-coupling capacitors, because the size of the form factor approximately matches the trace width commonly used for these signals.

The Enhanced SerDes interface can be directly connected to either 1 Gbps or 2.5 Gbps SFP modules. For these applications, external AC-coupling capacitors are not required, because the SFP module already includes capacitors.

The following table lists the AC-coupling requirements for common Enhanced SerDes connections.

***Table 913 •*** **Enhanced SerDes Interface Coupling Requirements**

| Enhanced SerDes Connection | Mode | External AC-Coupling Requirement |
|---|---|---|
| SFP modules | SFP | Not required |
| SGMII PHY | SGMII | Required(1) |
| Enhanced SerDes device | Enhanced SerDes | Required |

1. AC-coupling is not required with direct connection to the VSC8512 PHY device.

The Enhanced SerDes interface signals must be routed as a differential pair, with a 100 Ω differential characteristic impedance. The differential intrapair skew must be below 5 ps in the PCB trace.

To minimize crosstalk between transmitter and receiver, equal drive strength is recommended in both devices in a link.

To minimize crosstalk between differential pairs, the characteristic differential impedance of the signals must be determined only by the distance to the reference plane and the intra-pair coupling and not the distance to the neighboring traces. To separate the transmitter and receiver signals to minimize crosstalk, it is recommended to route the transmitter and receiver signals on as many different PCB layers as feasible.

## 14.4.6 Two-Wire Serial Interface

The two-wire serial interface is capable of suppressing small amplitude glitches less than 5 ns in duration, which is less than the 50 ns duration often quoted for similar interfaces. Because the two-wire serial implementation uses Schmitt-triggered inputs, the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices have a greater tolerance to low amplitude noise. For glitch-free operation, select the proper pull-up resistor value to ensure that the transition time through the input switching region is less than 5 ns given the line's total capacitive load. For capacitive loads up to 40 pF, a pull-up resistor of 510 Ω or less ensures glitch-free operation for noise levels up to 700 mV peak-to-peak.

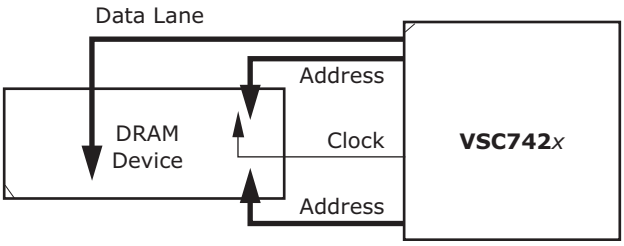
## 14.4.7 DDR2 SDRAM Interface

The DDR2 SDRAM interface is designed to interface directly with a single 8-bit DDR SDRAM device. The maximum supported density is 128 Mbyte (1 Gbps).

All signals on this interface must be connected one-to-one with the corresponding signals on the DDR SDRAM device. If the memory size of the DDR SDRAM is smaller than maximum, then the upper part of the address and bank address signals can be left unconnected. All eight data bits must be used.

The placement of the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 interface signals is optimized for point-to-point routing directly to a single DDR SDRAM device.

***Figure 115 •*** **DDR2 SDRAM Point-to-Point Routing**




Because reflections are absorbed by the driver, keep the physical distance of all the SDRAM interface signals below 1 ns to omit any external discrete termination on the address, command, control and clock lines.

When routing the DDR2 interface, attention must be paid to the skew, primary concern is skew within the byte lane between the differential strobe and the single-ended signals. Skew recommendations for the DDR2 interface are listed in the following table.

***Table 914 •*** **Recommended Skew Budget**

| Description | Signal | Maximum Skew |
|---|---|---|
| Skew within byte lane 0 | DDR_DQS/DDR_DQSn | 50 ps |
| Skew within address, command, and control bus | DDR_CK/DDR_CKn<br>DDR_nRAS<br>DDR_CKe<br>DDR_ODT<br>DDR_nCAS<br>DDR_nWE<br>DDR_BA[2:0]<br>DDR_A[13:0] | 100 ps |
| Skew between control bus clock and byte lane clock | DDR_CK/DDR_CKn<br>DDR_DQS/DDR_DQSn | 1250 ps |
| Control bus differential clock intrapair skew | DDR_CK/DDR_CKn | 5 ps |

- Use a shared voltage reference between the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 device's DDR_Vref supply and the DDR device's reference voltage.
- Generate the DDR_Vref from the $V_{DD_IODDR}$ supply using a resistor divider with value of 1 kΩ and an accuracy of 1% or better.
- Use a decoupling capacitance of at least 0.1 µF on the supply in a manner similar to $V_{DD_IODDR}$ and $V_{SS}$ to ensure tracking of supply variations; however, the time constant of the resistor divider and decoupling capacitance should not exceed the nReset assertion time after power on.

Recommend routing:

- DDR_CK/DDR_CKn must be routed as a differential pair with a 100 Ω differential characteristic impedance.
- DDR_DQS/DDR_DQSn must be routed as a differential pair with a 100 Ω differential characteristic impedance.
- To minimize crosstalk, the characteristic impedance of the single-ended signals should be determined predominantly by the distance to the reference plane and not the distance to the neighboring traces.
- The crosstalk should be below –20 dB.

## 14.4.8 Thermal Diode External Connection

The internal on-die thermal diode can be used with an external temperature monitor to easily and accurately measure the junction temperature of the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

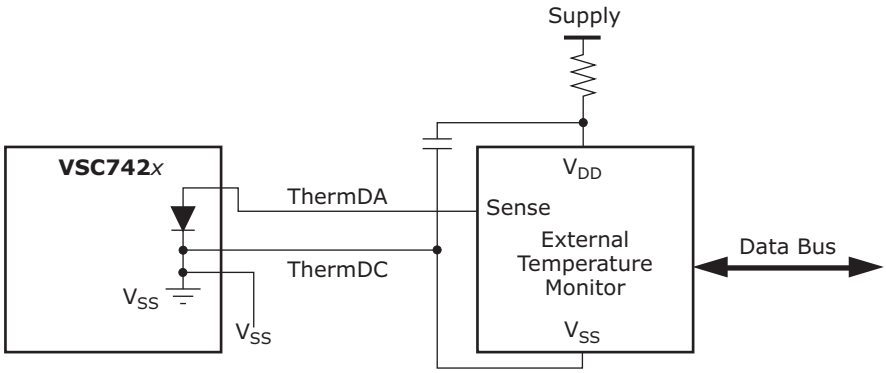
The on-die thermal diode has internal connected the diode cathode to $V_{SS}$, the external temperature sensor must support the thermal diode cathode connected to $V_{SS}$.

Thermal diode is extremely sensitive to noise. To minimize the temperature measurement errors, follow these guidelines:

- Route the ThermDC and ThermDA signals as a differential pair with a differential impedance less than 100 Ω.
- Place the external temperature monitor as close as is possible to the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.
- Add a 47 Ω resistor in series with the external temperature monitor supply to filter noise.
- Place a de-coupling capacitor between the external temperature monitor supply pin and the ThermDC signal. Place the capacitor close to the external temperature sensor, as shown in the following illustration.

Connect the external temperature monitor $V_{SS}$ pin directly to the ThermDC pin, which has the connection to $V_{SS}$, as shown in the following illustration. Do not connect the external temperature monitor $V_{SS}$ pin to the global $V_{SS}$ plane.

***Figure 116 •*** **External Temperature Monitor Connection**



# 15 Design Considerations

This section provides information about the design considerations for the VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 devices.

## 15.1 10BASE-T mode unable to re-establish link

10BASE-T mode is unable to re-establish link with the following devices if the link drops while sending data: SparX-III™ and Caracal™ family of switches, VSC8512-02, VSC8522-02, VSC8522-12, VSC8504, VSC8552, VSC8572, and VSC8574. No issue is observed for other link partner devices. The probability of this error occurring is low except in a test environment.

The workaround is to contact Microsemi for the current API software release.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100054.

## 15.2 Software script for link performance

Software script is required for improved link performance. PHY ports may exhibit suboptimal performance. Contact Microsemi for a script to be applied during system initialization.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100034.

## 15.3 10BASE-T signal amplitude

10BASE-T signal amplitude can be lower than the minimum specified in IEEE 802.3 paragraph 14.3.1.2.1 (2.2 V) at low supply voltages. This issue is not estimated to present any system level impact. Performance is not impaired with cables up to 130 m with various link partners.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100036.

## 15.4 Clause 45 register 7.60

Clause 45, register 7.60, bit 10 reads back as a logic 1. This is a reserved bit in the standard and should be ignored by software.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100037.

## 15.5 Clause 45 register 3.22

Clause 45, register 3.22 is cleared upon read only when extended page access register (register 31) is set to 0. This register cannot be read when page access register is set to a value other than 0.

The workaround is to set the extended page access register to 0 before accessing clause 45, register 3.22.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100038.

## 15.6 Clause 45 register 3.1

Clause 45, register 3.1, Rx and Tx LPI received bits are cleared upon read only when extended page access register (register 31) is set to 0.

The workaround is to set the extended page access register to 0 before accessing clause 45, register 3.1.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100039.

## 15.7 Clause 45 register address post-increment

Clause 45 register address post-increment only works when reading registers and only when the extended page access register (register 31) is set to 0. The estimated impact is low, as there are very few Clause 45 registers in a Gigabit PHY, and they can be addressed individually.

The workaround is to access Clause 45 registers individually.

This item was previously published in the *VSC7424-02, VSC7425-02, VSC7426-02, and VSC7427-02 Errata revision 1.0* as EA100040.

## 15.8 IEEE1588 Out of Sync Situation

For CuPHY port 10-11 and all Serdes ports with or without non timestamping PHY:

If a short frame of less than approximately 3 bytes is received on a port while the PCS-rx is enabled, the timestamp FIFO erroneously increments. This means that the timestamp of the previous packet is used in any IEEE1588 operation on the given port. The only way to bring the timestamp FIFO in sync is to do a full reset of the switch.

Work-around for CuPHY: Keep the PCS-rx disabled during link state changes to avoid illegal frames getting a timestamp that causes the OOS (out of sync) state.

### 15.8.1 Copper Port (internal CuPHY 10-11 and External PHYs Without Timestamping)

Initially, before link is up, the switch/port PCS-rx is disabled.

The PHY is configured to advertise all supported speeds (as configured for the port).

On link-up, software reads back the negotiated speed from the PHY and configures the MAC and then enables the PCS-rx.

On link-down, the PCS-rx is disabled.

When the link speed changes to 10M or 100M then the PHY autonegotiation capabilities are removed without restarting autonegotiation.

This is to avoid the PHY changing to a higher speed before the port PCS-rx is disabled.

Next time the link partner restarts autonegotiation the autonegotiation process will end up in not-resolved state with no change to the speed.

The software fix detects link down, disables the PCS-rx, restores the autonegotiation capabilities and restarts autonegotiation. When the new link speed is negotiated the PCS-rx is enabled.

The workaround requires change in the following.

- Port API to support PCS-rx enable/disable/ignore (New: "PCS" field in the vtss_port_conf_t struct).
- PHY API to support removal of autonegotiation capabilities. (New: "no_restart_aneg" member in vtss_phy_aneg_t struct).
- Application (the bulk of the fix).

Pseudo-code for disabling the PCS-rx during link changes:

```
Initialization
  Disable PCS-rx (see note);
  Aneg.cap = user_capabilities;

Port polling thread
  PHY status = no link;
    Disable PCS-rx;
    Aneg.cap = user_capabilities;
    Aneg restart;
  PHY status = link
```

```
    If Aneg.speed = 100Mbps or 10Mbps then Aneg.cap = none and Aneg no
    restart (see note);
    MAC.speed = PHY.speed
    Enable PCS-rx;
End port polling thread;

CLI thread (manual configuration)
  If Aneg.cap = 100Mbps or 10Mbps then {
  Disable PCS-rx;
  Aneg restart;}
End CLI thread
```

**Note:** Disable PCS-rx means setting bit DEV[port#]:PORT_MODE:CLOCK_CFG.PCS_RX_RST. In API, use the new "PCS" field in the vtss_port_conf_set()::vtss_port_conf_t::PCS to control the state of PCS-rx.

**Note:** Use the new member added to the vtss_phy_conf_set():: vtss_phy_aneg_t::no_restart_aneg.

A software patch for the application and API implementing this PCS-rx disable fix is available. Ensure your 1588-enabled software has this fix implemented. For information regarding official releases, check with your sales representative.

### 15.8.2 Serdes Port (SFP)

There is no way to prevent a short frame being received on a serdes port. Tests have shown that during disconnection of a fiber while the port is receiving frames at a high speed, many short frames (fragments) are received, and the OOS state is entered on the port.

New designs should use external timestamping PHY on serdes (SFP) ports. The switch port timestamping should not be used.

To avoid the OOS from occurring in existing designs, the port should be set to disabled from the management interface before removing the fiber. As this is not possible in all situations, implement a software work-around to examine the port for OOS state during link down and if OOS is detected, reset the switch with a log message stating OOS state was detected. To detect the OOS state, set the port in loop-back mode, send a PTP frame, and determine if the correct timestamp is used.

# 16 Ordering Information

VSC7424XJG-02, VSC7425XJG-02, VSC7426XJG-02, and VSC7427XJG-02 are packaged in a lead(Pb)-free, 672-pin, thermally enhanced, plastic ball grid array (BGA) with a 27 mm × 27 mm body size, 1 mm pin pitch, and 2.36 mm maximum height.

Lead-free products comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

The following table lists the ordering information.

***Table 916 •*** **Ordering Information: BGA Package**

| Part Order Number | Description |
|---|---|
| VSC7424XJG-02 | 10-port managed Gigabit Ethernet switch<br>Lead(Pb)-free, 672-pin, thermally enhanced, plastic BGA with a<br>27 mm × 27 mm body size, 1 mm pin pitch, and 2.36 mm maximum height |
| VSC7425XJG-02 | 18-port managed Gigabit Ethernet switch<br>Lead(Pb)-free, 672-pin, thermally enhanced, plastic BGA with a<br>27 mm × 27 mm body size, 1 mm pin pitch, and 2.36 mm maximum height |
| VSC7426XJG-02 | 24-port managed Gigabit Ethernet switch<br>Lead(Pb)-free, 672-pin, thermally enhanced, plastic BGA with a<br>27 mm × 27 mm body size, 1 mm pin pitch, and 2.36 mm maximum height |
| VSC7427XJG-02 | 26-port managed Gigabit Ethernet switch<br>Lead(Pb)-free, 672-pin, thermally enhanced, plastic BGA with a<br>27 mm × 27 mm body size, 1 mm pin pitch, and 2.36 mm maximum height |