

88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU
System on Chip

Functional Specifications – Unrestricted

Doc. No. MV-S109094-U0, Rev. A
November 25, 2015, Preliminary

Document Classification: Proprietary Information

Document Conventions

	**Note:** Provides related information or information of special importance.
	**Caution:** Indicates potential damage to hardware or software, or loss of data.
	**Warning:** Indicates a risk of personal injury.

Document Status

Doc Status: Preliminary	Technical Publication: 0.xx

For more information, visit our website at: http://www.marvell.com

Disclaimer

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Marvell. Marvell retains the right to make changes to this document at any time, without notice. Marvell makes no warranty of any kind, expressed or implied, with regard to any information contained in this document, including, but not limited to, the implied warranties of merchantability or fitness for any particular purpose. Further, Marvell does not warrant the accuracy or completeness of the information, text, graphics, or other items contained within this document.

Marvell products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use Marvell products in these types of equipment or applications.

With respect to the products described herein, the user or recipient, in the absence of appropriate U.S. government authorization, agrees:

1) Not to re-export or release any such information consisting of technology, software or source code controlled for national security reasons by the U.S. Export Control Regulations ("EAR"), to a national of EAR Country Groups D:1 or E:2;

2) Not to export the direct product of such technology or such software, to EAR Country Groups D:1 or E:2, if such technology or software and direct products thereof are controlled for national security reasons by the EAR; and,

3) In the case of technology controlled for national security reasons under the EAR where the direct product of the technology is a complete plant or component of a plant, not to export to EAR Country Groups D:1 or E:2 the direct product of the plant or major component thereof, if such direct product is controlled for national security reasons by the EAR, or is subject to controls under the U.S. Munitions List ("USML").

At all times hereunder, the recipient of any such information agrees that they shall be deemed to have manually signed this document in connection with their receipt of any such information.

Copyright © 1999-2015. Marvell International Ltd. All rights reserved. Alaska, ARMADA, Avanta, Avastar, CarrierSpan, Kinoma, Link Street, LinkCrypt, Marvell logo, Marvell, Moving Forward Faster, Marvell Smart, PISC, Prestera, Qdeo, QDEO logo, QuietVideo, Virtual Cable Tester, The World as YOU See It, Vmeta, Xelerated, and Yukon are registered trademarks of Marvell or its affiliates. G.now, HyperDuo, Kirkwood, and Wirespeed by Design are trademarks of Marvell or its affiliates.

Patent(s) Pending-Products identified in this document may be covered by one or more Marvell patents and/or patent applications.

Revision History

Revision	Date	Description
A	November 25, 2015	Unrestricted Release

Copyright © 2015 Marvell

Table of Contents

Revision History **3**

Preface **19**

About this Document 19
Document Organization 19
Related Documents 20
Document Conventions 21

1 ARMADA® 38x Family Overview **25**

2 Detailed Features List **27**
2.1 Device Differences 34

3 Typical Applications and System Configurations **37**
3.1 Small/Medium Business (SMB) NAS 37
3.2 Enterprise Wireless Access Point (WAP) 38
3.3 Smart Gateway 39
3.4 Network Video Recorder (NVR) 40
3.5 Main CPU in a Control Plane 41

4 Address Map **43**
4.1 Internal Address Map 43
4.2 Cortex™-A9 Processor Address Decoding 45
4.3 PCI Express Address Decoding 51
4.4 XOR DMA Address Decoding 55
4.5 USB Address Decoding 55
4.6 SATA Address Decoding 56
4.7 Security Accelerator Address Decoding 56
4.8 SDIO Address Decoding 56
4.9 TDM Address Decoding 56
4.10 I2C Address Decoding 56
4.11 Audio Address Decoding 57

5 System Considerations **58**
5.1 Data Integrity 58
5.2 PCI Express Reference Clock Output 63
5.3 Big and Little Endian Support 64
5.4 Spread Spectrum Clock Generator (SSCG) 65
5.5 Integrated Thermal Sensor 66

Copyright © 2015 Marvell

6 Boot Flow 67
6.1 Features 67

7 CPU Subsystem 71
7.1 CPU Subsystem Features 73
7.2 Interrupt Controller 74
7.3 I/O Coherency 94

8 Timers, Counters, and Watchdog 95
8.1 Features 96
8.2 Functional Description 96

9 DRAM Memory Controller 101
9.1 Feature List 101
9.2 DDR4 Features 103

10 NAND Flash Controller (NFC) 105
10.1 Features 106
10.2 NAND Flash Interface 107
10.3 NAND Flash Connectivity 107
10.4 Operation 111
10.5 Usage Models 127

11 Device Bus Controller 141
11.1 Features 142
11.2 Functional Description 142
11.3 Address Multiplexing 145
11.4 Device Bus Timing 147
11.5 Data Pack/Unpack and Burst Support 154
11.6 READYn Support 154
11.7 Additional Device Interface Signaling 156

12 Ethernet Networking Controller 159
12.1 Networking Controller Modes Overview 162
12.2 Functional Description 164
12.3 Queues Management – Counters Mode 169

13 1G/2.5G Ethernet MAC 183
13.1 Features 183
13.2 Functional Description 183
13.3 Serial Management Interface (SMI)—Ethernet MAC 184
13.4 Port MIB Counters 184

Copyright © 2015 Marvell

14 Precise Time Protocol (PTP) 189

15 Hardware Buffer Management Controller 190
15.1 Features 190

16 Time Division Multiplexing (TDM) Controller 193
16.1 Features 193
16.2 Functional Description 194
16.3 TDM Protocol Specification 195

17 PCI Express Interface (PCIe) 2.0 207
17.1 Features 207
17.2 Functional Description 208
17.3 Link Initialization 210
17.4 Disabling a Link 210
17.5 Master Memory Transactions 210
17.6 Master I/O Transactions 211
17.7 Master Configuration Transactions 211
17.8 Target Memory Transactions 212
17.9 Target I/O Transactions 212
17.10 Target Configuration Transactions 212
17.11 Target Special Cases 213
17.12 Messages 213
17.13 Message Signaled Interrupt (MSI) Support 215
17.14 Locked Transactions 215
17.15 Arbitration and Ordering 215
17.16 PCI Express Register Access 216
17.17 Hot Reset 216
17.18 Error Handling 217
17.19 Power Management 220
17.20 Loopback Modes 221
17.21 Peer-to-Peer Traffic 223

18 Universal Serial Bus (USB) Controller 224
18.1 Universal Serial Bus (USB 2.0) Interface 225
18.2 Universal Serial Bus (USB) 3.0 Host Controller 227
18.3 Universal Serial Bus (USB) 3.0 Device Controller 229

19 Serial Advanced Technology Attachment (SATA) 3.0 Host Controller 240
19.1 Features 241
19.2 High-Level Functional Description—SATA Port Access 241
19.3 AHCI Mode 244
19.4 Configuring Amplitude and Pre-emphasize Level 249

Document Classification: Proprietary Information

20 Audio (I^2S / S/PDIF) Interface 250
20.1 Features 251
20.2 Recording Data Flow 253
20.3 Playback Flow 256
20.4 Error Handling 261
20.5 Audio Unit Memory Structure 261

21 Serial Peripheral Interface (SPI) 265
21.1 Features 266
21.2 Functional Description 266
21.3 SPI Modes 267
21.4 Indirect Access 269
21.5 Direct Mode 270

22 Secure Digital Input/Output (SDIO) 3.0 274
22.1 Features 275
22.2 MMC Mode 276
22.3 Programming Guidelines 276
22.4 SD Clock Selection 278
22.5 Power Up Sequence 278

23 Inter-Integrated Circuit Interface (I^2C) 279
23.1 Features 280
23.2 I^2C Bus Operation 280
23.3 I^2C Functional Description 281
23.4 I^2C Serial ROM Initialization 289
23.5 I^2C Hardware Slave 289

24 Universal Asynchronous Receiver/Transmitter (UART) Interface 291
24.1 Features 292
24.2 Functional Description 293
24.3 Programmable Baud-Rate Generator 294
24.4 UART Debug Mode 295

25 General Purpose Input/Output Ports 296
25.1 Features 296
25.2 Functional Description 296
25.3 LEDs and Pulse Width Modulation (PWM) 296
25.4 GPIO Interrupts 297
25.5 Data Input Sampling 297
25.6 Data Output Driving 297
25.7 Multi-CPU Support 298

Copyright © 2015 Marvell

26 Real-Time Clock (RTC) 303
26.1 Features 303
26.2 Status and Control Registers Description 304
26.3 Seconds to Time Conversion 305
26.4 Clock Correction Implementation 306

27 Cryptographic Engines and Security Accelerators (CESA) 308
27.1 Features 309
27.2 Theory of Operation 310

28 Power Management Controller 311
28.1 Features 312
28.2 Functional Description 312
28.3 Individual CPU Power Management 312
28.4 Dynamic Frequency Scaling (DFS) 315
28.5 Waking Options 316
28.6 Thermal Management 317
28.7 Idle Profiling Counter 317
28.8 Unit Specific Power Management Options 318
28.9 Debug During Power Management (CoreSight) 318
28.10 Power Management Registers 318

29 High-Speed Integrated SERDES Interface 320
29.1 Features 320
29.2 SERDES Configuration 320
29.3 Assigning a SERDES Lane to the Relevant MAC 322
29.4 Configuring Amplitude and Pre-emphasize Level 328

30 XOR Engines 329
30.1 Features 329
30.2 Functional Description 330
30.3 XOR Engine Modes of Operation 331
30.4 Descriptor Chain Format 335
30.5 Address Decoding 343
30.6 Arbitration 344
30.7 XOR Engine Programming 345
30.8 Burst Limit 350
30.9 Errors and Interrupts 350
30.10 Performance Optimization 351

31 Electronic Fuse (eFuse) 352
31.1 Features 352

Copyright © 2015 Marvell

A ARMADA® 38x Family Register Set 438
A.1 Registers Overview 438
A.2 CPU Subsystem Registers 439
A.3 CPU Power Management Unit Registers 508
A.4 NAND Flash Registers 517
A.5 Device Bus Controller Registers 544
A.6 Time Division Multiplexing Registers 552
A.7 PCI Express 2.0 Port 0 Registers 571
A.8 PCI Express 2.0 Ports 1-3 Registers 628
A.9 USB 2.0 Registers 691
A.10 USB2 PHY (UTMI PHY) Registers 698
A.11 USB 3.0 Host Controller Registers 717
A.12 USB 3.0 Host Controller Bridge Registers 745
A.13 USB 3.0 Device Registers 748
A.14 SATA 3.0 Host Controller Registers 1027
A.15 Audio (I2S / S/PDIF) Registers 1054
A.16 Serial Peripheral Interface (SPI) Registers 1073
A.17 Secure Digital Input/Output (SDIO) 3.0 Registers 1082
A.18 Inter-Integrated Circuit (I2C) Registers 1120
A.19 Universal Asynchronous Receiver/Transmitter (UART) Registers 1130
A.20 General Purpose Input/Output (GPIO) Registers 1140
A.21 Real-Time Clock (RTC) Registers 1150
A.22 Power Management Unit Registers 1153
A.23 Common PHY Registers 1155
A.24 Common PHY and PIPE Interface Registers 1170
A.25 XOR Engine Registers 1363
A.26 Multi-Purpose Ports (MPP) Registers 1382
A.27 General Management - DFX Server Registers 1384
A.28 Clock Complex Registers 1392
A.29 System Registers 1393

B Acronyms and Abbreviations 1423

Copyright © 2015 Marvell

List of Tables

Revision History **3**

Preface **19**

1 ARMADA® 38x Family Overview **25**

2 Detailed Features List **27**

Table 1: Device Differences 34

3 Typical Applications and System Configurations **37**

4 Address Map **43**

Table 2: Internal Address Map 43
Table 3: CPU Interface Mbus Decoding Units IDs and Attributes 46
Table 4: Cortex™-A9 Processor Default Address Map 48
Table 5: Unit IDs and Attributes 52

5 System Considerations **58**

Table 6: ECC Code Matrix 58

6 Boot Flow **67**

7 CPU Subsystem **71**

Table 7: Distributer Interrupt Map 78
Table 8: Proprietary Interrupt Controller Interrupts 86
Table 9: Proprietary Interrupt Controller Errors 91

8 Timers, Counters, and Watchdog **95**

9 DRAM Memory Controller **101**

10 NAND Flash Controller (NFC) **105**

Table 10: NAND Attribute Definitions 106
Table 11: CE Assignment 108
Table 12: Spare Area Used for ECC 112
Table 13: Even Data Stream 113
Table 14: Odd Data Stream 113
Table 15: ECC Byte Placement 114
Table 16: Small Block Page Allocation 116
Table 17: Large Block Page Allocation 116
Table 18: 4-KB Physical or Logical Page Allocation Using 2-KB Chunks 118
Table 19: Read 2 KB MLC Page 122
Table 20: Read 4 KB MLC Page 123
Table 21: Program (Write) 2 KB MLC Page 124

Copyright © 2015 Marvell

Table 22: Program (Write) 4 KB MLC Page 125

11 Device Bus Controller 141

Table 23: Read Timing Parameters 151

Table 24: Write Timing Parameters 152

12 Ethernet Networking Controller 159

Table 25: Coding for <AccelerationMode> 162

Table 26: Transmit Descriptor Description 171

Table 27: Transmit Descriptor—Offset 0 172

Table 28: Transmit Descriptor—Offset 4 173

Table 29: Transmit Descriptor—Offset 8 174

Table 30: Transmit Descriptor—Offset C 174

Table 31: Transmit Descriptor—Offset 10 175

Table 32: Transmit Descriptor—Offset 14 175

Table 33: Transmit Descriptor—Offset 18 175

Table 34: Transmit Descriptor—Offset 1C 175

Table 35: Receive Descriptor – Description 178

Table 36: Receive Descriptor—Offset 0 178

Table 37: Receive Descriptor—Offset 4 180

Table 38: Receive Descriptor—Offset 8 181

Table 39: Receive Descriptor—Offset C 181

Table 40: Receive Descriptor—Offset 10 181

Table 41: Receive Descriptor—Offset 14 182

Table 42: Receive Descriptor—Offset 18 182

13 1G/2.5G Ethernet MAC 183

Table 43: Definitions for MAC MIB Counters 185

14 Precise Time Protocol (PTP) 189

15 Hardware Buffer Management Controller 190

16 Time Division Multiplexing (TDM) Controller 193

Table 44: Time Division Multiplexing (TDM) Interface Signals 195

17 PCI Express Interface (PCIe) 2.0 207

Table 45: Supported Message Groups—Root Complex Mode 213

Table 46: Supported Message Groups—Endpoint Mode 214

Table 47: Physical Layer Error List 217

Table 48: Data Link Layer Error List 217

Table 49: Transaction Layer Error List 218

18 Universal Serial Bus (USB) Controller 224

Table 50: Reset on Device SuperSpeed Logic Module and State 231

Table 51: Buffer Descriptor Structure 234

Table 52: Buffer Descriptor Field Descriptions 234

Copyright © 2015 Marvell

Table 53: DONEQ Field Descriptions 236
Table 54: SuperSpeed Link Power States 239
Table 55: USB2.0 Link Power States 239

19 Serial Advanced Technology Attachment (SATA) 3.0 Host Controller 240
Table 56: SATA Received FIS Structure 245
Table 57: SATA Command List Structure 245
Table 58: SATA Command Header Structure 246
Table 59: SATA Command Header DW0 246
Table 60: SATA Command Header DW1 247
Table 61: SATA Command Header DW2 247
Table 62: SATA Command Header DW3 248
Table 63: SATA Command Table Content 248
Table 64: Physical Region Descriptor Table Command Structure 248
Table 65: Physical Region Descriptor—DW0–DW3 249

20 Audio (I^2S / S/PDIF) Interface 250
Table 66: Audio Unit Memory Bit Description 264

21 Serial Peripheral Interface (SPI) 265

22 Secure Digital Input/Output (SDIO) 3.0 274
Table 67: SD Feature Support 275

23 Inter-Integrated Circuit Interface (I^2C) 279
Table 68: I^2C Control Register Bits 282
Table 69: I^2C Status Codes 283

24 Universal Asynchronous Receiver/Transmitter (UART) Interface 291
Table 70: Typical Baud Rates, TCLK = 200 MHz 294
Table 71: Typical Baud Rates, TCLK = 250 MHz 295

25 General Purpose Input/Output Ports 296

26 Real-Time Clock (RTC) 303
Table 72: Clock Correction Mode Specifications 306

27 Cryptographic Engines and Security Accelerators (CESA) 308

28 Power Management Controller 311
Table 73: CPU Core Power Modes 312
Table 74: CPU Core Power Management Register Pointers 318

29 High-Speed Integrated SERDES Interface 320
Table 75: PCIe PIPE Configurations 324
Table 76: PHY MODE and REF_FREF_SEL Setting 324
Table 77: SEL_BITS Settings 325
Table 78: REFCLK_SEL Setting 325

Copyright © 2015 Marvell

Table 79: PIN_PHY_GEN Setting 326
Table 80: PCIe Pipe Configurations 326
Table 81: Completion Indications 327

30 XOR Engines 329
Table 82: Descriptor Status Word Definition 337
Table 83: Descriptor CRC-32 Result Word Definition 337
Table 84: Descriptor Command Word Definition 337
Table 85: Descriptor Next Descriptor Address Word 341
Table 86: Descriptor Byte Count Word 342
Table 87: Descriptor Destination Address Word 342
Table 88: Descriptor Source Address #N Words 342
Table 89: Descriptor Q Buffer Address Word 342
Table 90: EOC/EOD interpretation 350

31 Electronic Fuse (eFuse) 352

A ARMADA® 38x Family Register Set 438
Table 91: Standard Register Field Type Codes 438
Table 92: Summary Map Table for the CPU Subsystem Registers 439
Table 268: Summary Map Table for the CPU Power Management Unit Registers 508
Table 289: Summary Map Table for the NAND Flash Registers 517
Table 303: Summary Map Table for the Device Bus Controller Registers 544
Table 314: Summary Map Table for the Time Division Multiplexing Registers 552
Table 354: Summary Map Table for the PCI Express 2.0 Port 0 Registers 571
Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers 628
Table 544: Summary Map Table for the USB 2.0 Registers 691
Table 560: Summary Map Table for the USB2 PHY (UTMI PHY) Registers Registers 698
Table 576: Summary Map Table for the USB 3.0 Host Controller Registers 717
Table 644: Summary Map Table for the USB 3.0 Host Controller Bridge Registers 745
Table 650: Summary Map Table for the USB 3.0 Device Registers 748
Table 1250: Summary Map Table for the SATA 3.0 Host Controller Registers 1027
Table 1292: Summary Map Table for the Audio (I2S / S/PDIF) Registers 1054
Table 1325: Summary Map Table for the Serial Peripheral Interface (SPI) Registers 1073
Table 1340: Summary Map Table for the Secure Digital Input/Output (SDIO) 3.0 Registers 1082
Table 1382: Summary Map Table for the Inter-Integrated Circuit (I2C) Registers 1120
Table 1401: Summary Map Table for the Universal Asynchronous Receiver/Transmitter (UART) Registers 1130
Table 1417: Summary Map Table for the General Purpose Input/Output (GPIO) Registers 1140
Table 1454: Summary Map Table for the Real-Time Clock (RTC) Registers 1150
Table 1463: Summary Map Table for the Power Management Unit Registers 1153
Table 1469: Summary Map Table for the Common PHY Registers 1155
Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers 1170
Table 1838: Summary Map Table for the XOR Engine Registers 1363
Table 1862: Summary Map Table for the Multi-Purpose Ports (MPP) Registers 1382
Table 1865: Summary Map Table for the General Management – DFX Server Registers 1384

Copyright © 2015 Marvell

Table 1883: Summary Map Table for the Clock Complex Registers 1392
Table 1886: Summary Map Table for the System Registers 1393

B Acronyms and Abbreviations 1423

Copyright © 2015 Marvell

List of Figures

1 ARMADA® 38x Family Overview 25
Figure 1: ARMADA® 380, 385 and 388 Block Diagram 25
Figure 2: ARMADA® 381, 382 and 383 Block Diagram 26

2 Detailed Features List 27

3 Typical Applications and System Configurations 37
Figure 3: SMB NAS Application 37
Figure 4: Wi-Fi Access Point Application 38
Figure 5: Smart Gateway Application 39
Figure 6: NVR Application 40
Figure 7: Main CPU in a Control Plane Application 41
Figure 8: Line Card Management Application 42

4 Address Map 43

5 System Considerations 58

6 Boot Flow 67

7 CPU Subsystem 71
Figure 9: Processor Subsystem Block Diagram 72
Figure 10: Multiprocessor Interrupt Controller Flow 75
Figure 11: MSI/MSI-X Capturing Scheme Using the Software Trigger Interrupt 78

8 Timers, Counters, and Watchdog 95
Figure 12: Dual-CPU SoC Timers Topology 95

9 DRAM Memory Controller 101

10 NAND Flash Controller (NFC) 105
Figure 13: NAND Flash Controller Block Diagram 105
Figure 14: Stacked Flash System Example Using ND_CEn[0] and ND_CEn[1] 108
Figure 15: Striped Flash System Example Using ND_CEn[0] and ND_CEn[1] 109
Figure 16: Stacked Flash System Example Using ND_CEn[0] and ND_CEn[2] 110
Figure 17: ECC Code Generation 113
Figure 18: Hamming Error Detection Process 114
Figure 19: Erase Semantic Flowchart 120
Figure 20: Reset Semantic Flowchart 121
Figure 21: tWC Timing Example 128
Figure 22: tRC Timing Example Using Cnt_Del 130
Figure 23: tCS Timing Example 131

Copyright © 2015 Marvell

Figure 24: First Command Event Causality Diagram 133
Figure 25: Reset Command Event Causality Diagram 135
Figure 26: Read ID Command Event Causality Diagram 136
Figure 27: Write Command Event Causality Diagram 138

11 Device Bus Controller 141
Figure 28: Device Bus Controller Block Diagram 141
Figure 29: Device Bus Connectivity for up to 256 MB Address Space ..... (Single External Latch)—OEn/WEn in Shared Mode 143
Figure 30: Device Bus Connectivity (2 External Latches)—OEn/WEn in Non-Shared Mode 144
Figure 31: Address Multiplexing with 1 Latch and OEn/WEn in Shared Mode 145
Figure 32: Address Multiplexing with 2 Latches and OEn/WEn in Non-Shared Mode 146
Figure 33: Read Parameters Example 149
Figure 34: Write Parameters Example 150
Figure 35: NOR Interface Read Timing Diagram 151
Figure 36: NOR Interface Write Timing Diagram 152
Figure 37: READYn Extending Acc2First Example 155
Figure 38: READYn Extending Acc2Next Example 155
Figure 39: READYn Extending WrLow Example 155
Figure 40: BURSTn/DEV_LASTn Example 156

12 Ethernet Networking Controller 159
Figure 41: Ethernet Networking Controller Block Diagram 159
Figure 42: Ethernet Networking Controller Transmit Packet Walkthrough 160
Figure 43: Ethernet Networking Controller Receive Packet Walkthrough 161
Figure 44: Descriptors Management by Counters Transmit Queue 170
Figure 45: Descriptors Management by Counters Receive Queue 176

13 1G/2.5G Ethernet MAC 183
Figure 46: Ethernet Frame Classification 186
Figure 47: Bad Frame Procedure 187

14 Precise Time Protocol (PTP) 189

15 Hardware Buffer Management Controller 190

16 Time Division Multiplexing (TDM) Controller 193
Figure 48: SLIC/SLAC/DAA/Codec Connection Example 193
Figure 49: TDM Controller Block Diagram 194
Figure 50: TDM Operation Time Slot 0 196
Figure 51: TDM Wideband Mode Operation 197
Figure 52: TDM Transmit Path 199
Figure 53: TDM Receive Path 203

17 PCI Express Interface (PCIe) 2.0 207
Figure 54: High-level Block Diagram 207

Copyright © 2015 Marvell

Figure 55: Shallow Internal Loopback....222
Figure 56: Deep Internal Loopback....222

18 Universal Serial Bus (USB) Controller....224
Figure 57: USB 3.0 Host and Device Controller Block Diagram....224
Figure 58: USB 2.0 Port Block Diagram....225
Figure 59: USB 3.0 Host Top Level Diagram....228
Figure 60: USB 3.0 Device Top Level Diagram....230
Figure 61: Device Port Connection State Machine....238

19 Serial Advanced Technology Attachment (SATA) 3.0 Host Controller....240
Figure 62: SATAHC Block Diagram....240

20 Audio (I^2S / S/PDIF) Interface....250
Figure 63: Audio Unit Block Diagram....251
Figure 64: Recording Flow....255
Figure 65: Playback Flow....259
Figure 66: Memory Structure for Transmit and Receive....263

21 Serial Peripheral Interface (SPI)....265
Figure 67: SPI Controller Block Diagram....265
Figure 68: SPI Functional Diagram—Direct Access....266
Figure 69: SPI Functional Diagram—Indirect Access....267
Figure 70: SPI Timing for CPHA=0....268
Figure 71: SPI Timing for CPHA=1....268
Figure 72: SPI Timing Example....273

22 Secure Digital Input/Output (SDIO) 3.0....274
Figure 73: SDIO Host Block Diagram....274
Figure 74: SD Clock Multiplexing Structure for Clock Selection (sd_clk_fb)....278

23 Inter-Integrated Circuit Interface (I^2C)....279
Figure 75: I^2C Transaction Flow Diagram....279
Figure 76: I^2C Examples....281
Figure 77: I^2C Debug Port Write Example....289
Figure 78: I^2C Debug Port Read Example....290

24 Universal Asynchronous Receiver/Transmitter (UART) Interface....291
Figure 79: UART Character Format....291
Figure 80: UART Controller Block Diagram....292
Figure 81: UART Character (Two Stop Bits Example)....293
Figure 82: UART Character (One Stop Bit Example)....293

Copyright © 2015 Marvell

25 General Purpose Input/Output Ports 296

26 Real-Time Clock (RTC) 303
Figure 83: RTC Block Diagram 303

27 Cryptographic Engines and Security Accelerators (CESA) 308
Figure 84: CESA Functional Block Diagram 308

28 Power Management Controller 311
Figure 85: Power Management Block Diagram 311

29 High-Speed Integrated SERDES Interface 320
Figure 86: Assigning SERDES to a MAC 322

30 XOR Engines 329
Figure 87: XOR Engine Block Diagram 329
Figure 88: XOR Operation with Multiple Incoming Data Blocks 331
Figure 89: RAID6 Generation Flow Diagram 332
Figure 90: XOR iSCSI CRC32C Operation 333
Figure 91: XOR Descriptor Format 336
Figure 92: Programmable Channel Pizza Arbiter 344
Figure 93: Software and Hardware Synchronization 348

31 Electronic Fuse (eFuse) 352

A ARMADA® 38x Family Register Set 438

B Acronyms and Abbreviations 1423

Copyright © 2015 Marvell

Preface

About this Document

This document describes the architectural aspects and features implemented in the Marvell® ARMADA® 38x Family of devices. This document also provides detailed definitions of the registers implemented in these devices.

All feature descriptions and specifications described in this document refer to all devices, unless otherwise specified. In this document, the 88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828 are often referred to as the “device” or “devices”.

Relevant Devices

ARMADA 380	88F6810
ARMADA 381	88F6811
ARMADA 382	88F6821
ARMADA 383	88F6W21
ARMADA 385	88F6820
ARMADA 388	88F6828

Document Organization

This document is divided into the following parts an appendices:

- Part 1: Overview on page 23
- Part 2: CPU Subsystem on page 69
- Part 3: External Memory Interfaces on page 99
- Part 4: Ethernet Networking Subsystem on page 157
- Part 5: External Interfaces on page 191
- Part 6: System Functions and Engines on page 301
- Appendix A: Register Set on page 353

The sections of this document are organized according to architectural and functional topics.

Each section includes a description of the particular functional behavior or interface, and corresponds to an associated register appendix.

Copyright © 2015 Marvell

Related Documents

The following documents contain additional information related to the ARMADA® 38x Family of devices.

Title	Document Number
88F6810, 88F6820, and 88F6828 Hardware Specifications	MV-S108817-00
88F6811, 88F6821, and 88F6W21 Hardware Specifications	MV-S110251-00
88F6810, 88F6820, and 88F6828 ARMADA® 38x Family Hardware Design Guide	MV-S302310-00
88F6811, 88F6821, 88F6W21 ARMADA® 381, 382, and 383 High-Performance Single/Dual Core CPU System on Chip Hardware Design Guide	MV-S302453-00
External Standards	
Cortex-A9 Technical Reference Manual	ARM DDI 0338
Cortex-A9™ MPCore Technical Reference Manual	ARM DDI 0407
Cortex-A9 Floating-Point Unit Technical Reference Manual	ARM DDI 0408
Cortex-A9 NEON™ Media Processing Engine Technical Reference Manual	ARM DDI 0409
Corelink™ Level 2 Cache Controller L2C-310 Technical Reference Manual	ARM DDI 0246
ARM Generic Interrupt Controller Architecture Specification, Version 1.0	ARM IHI0048B
Cortex™-A9 MBIST Controller Technical Reference Manual	ARM DDI0414
ARM Architecture Reference Manual, Second Edition	
Serial ATA Revision 3.0 Specification	
Universal Serial Bus Revision 3.0 Specification	
eXtensible Host Controller Interface for Universal Serial Bus (xHCI) Specification, Revision 1.0	
JEDEC DDR3 SDRAM Standard	
JEDEC DDR4 SDRAM Standard	

For the latest product documentation, visit the Marvell Extranet.

Copyright © 2015 Marvell

Document Conventions

The following conventions are used in this document:

Signal Range	A signal name followed by a range enclosed in brackets represents a range of logically related signals. The first number in the range indicates the most significant bit (MSb) and the last number indicates the least significant bit (LSb). Example: DB_Addr[12:0]
Active Low Signals #	An n letter at the end of a signal name indicates that the signal's active state occurs when voltage is low. Example: INTn
State Names	State names are indicated in *italic* font. Example: *linkfail*
Register Naming Conventions	Register field names may be indicated by angle brackets. Example: <RegInit> Register field bits are enclosed in brackets. Example: Field [1:0] Register addresses are represented in hexadecimal format. Example: 0x0 Reserved: The contents of the register are reserved for internal use only or for future use. A lowercase <n> in angle brackets in a register indicates that there are multiple registers with this name. Example: Multicast Configuration Register<n>
Reset Values	Reset values have the following meanings: 0 = Bit clear 1 = Bit set
Abbreviations	Kb: kilobit KB: kilobyte Mb: megabit MB: megabyte Gb: gigabit GB: gigabyte
Numbering Conventions	Unless otherwise indicated, all numbers in this document are decimal (base 10). An 0x prefix indicates a hexadecimal number. An 0b prefix indicates a binary number.

Copyright © 2015 Marvell

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU System on Chip

Functional Specifications – Unrestricted

Part 1: Overview

- Section 1, ARMADA® 38x Family Overview
- Section 2, Detailed Features List
- Section 3, Typical Applications and System Configurations
- Section 4, Address Map
- Section 5, System Considerations
- Section 6, Boot Flow

Doc. No. MV-S109094-U0, Rev. A
November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell

1 ARMADA® 38x Family Overview

This section presents the block diagrams of the Marvell® ARMADA® 38x Family of devices.

Figure 1: ARMADA® 380, 385 and 388 Block Diagram

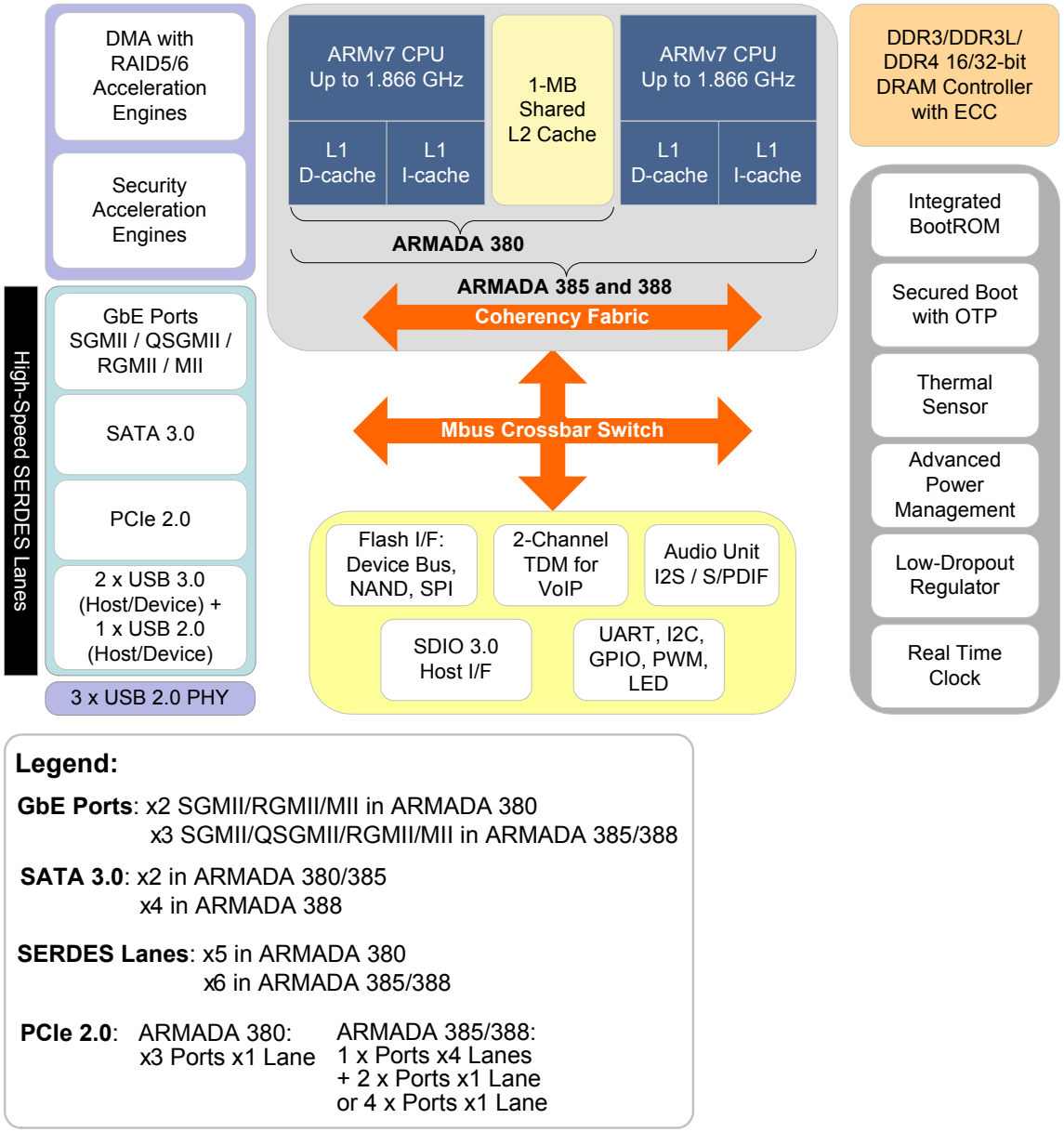
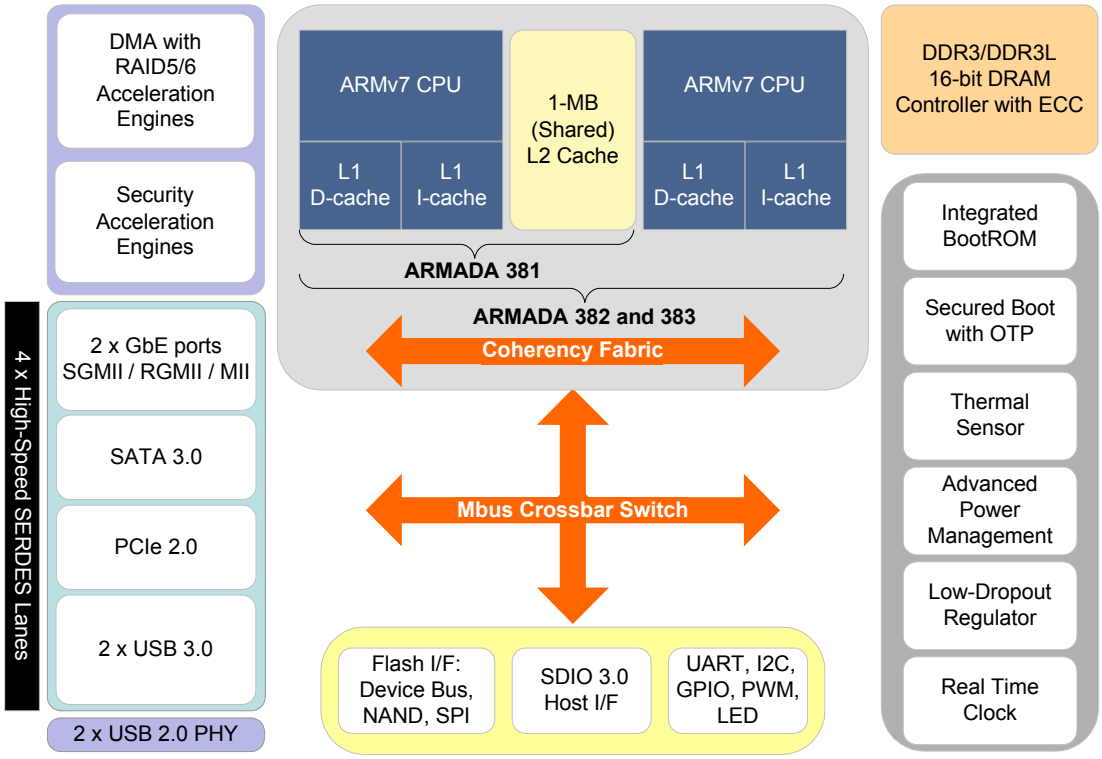


Figure 2: ARMADA® 381, 382 and 383 Block Diagram



Legend:

CPU: Up to 1.33 GHz in ARMADA 381/382
Up to 1.2 GHz in ARMADA 383

SATA 3.0: x2 in ARMADA 381/382
x1 in ARMADA 383

PCIe 2.0: x3 in ARMADA 381/382
x2 in ARMADA 383

USB 3.0: Host/Device in ARMADA 381/382
Host Only in ARMADA 383

Copyright © 2015 Marvell

2 Detailed Features List

ARMv7 Cortex-A9 CPU(s)

The ARMv7 core supports the following features:

- **88F6810/88F6820/88F6828**—Up to 1.866 GHz
- **88F6811/88F6821**—Up to 1.33 GHz CPU clock
- **88F6W21**—Up to 1.2 GHz CPU clock
- Superscalar RISC CPU with Harvard architecture, variable length, out-of-order pipeline
- NEON SIMD co-processor for improved signal processing
- DSP instructions to boost performance for multimedia and signal processing algorithms
- 32-bit instruction set for high-performance and flexibility
- Thumb-2 and Thumb-EE instruction set for code density
- Single / double precision Floating Point Unit (FPU)
- Symmetrical Multiprocessing (SMP) and Asymmetric Multiprocessing (AMP) modes
- 32-KB L1 Instruction cache 4-way, set-associative, virtually indexed, physically tagged, parity protected
- 32-KB L1 Data cache, 4-way, set-associative, physically indexed, physically tagged, parity protected
- L1 Prefetch:
  - Software driven using ARM PLD instructions
  - Hardware prefetcher that monitors up to 2 data streams
- Table Look-ahead Buffer (TLB):
  - Micro TLB: Two 64-entry fully-associative for Instruction and Data caches
  - Macro TLB: 512-entry 2-way associative structure
- MESI cache coherency scheme
- Hit-under-miss and multiple outstanding requests
- 8-stage pipeline
- Out-of-order execution for increased performance
- Branch Prediction Unit:
  - 2-way Branch Target Address Cache (BTAC) of 4096 entries
  - Global History Buffer (GHB) with 16K 2-bit predictors
  - Return Stack with eight 32-bit entries
- 64-bit internal data bus with 64-bit load / store instructions
- Endianess options—Little or Mixed Endian
- Performance Monitor Unit (PMU) that counts up to 6 event types
- JTAG/ARM-compatible ICE and Embedded Trace Module (ETM) for enhanced real-time debug capabilities

Copyright © 2015 Marvell

Unified Layer 2 Cache

- 16-way, 1024 KB, write-back and write-through cache
- Physically Indexed Physically Tagged (PIPT)
- Non-blocking pipeline supports multiple outstanding requests and Hit Under Miss (HUM) operation
- Lockdown/way-locking for data and instructions by line or master ID
- Parity protected

DDR3 / DDR3L / DDR4 SDRAM Controller

- 16/32-bit, with an Error-Correcting Code (ECC) option supported by both interface widths
- **88F6810/88F6820/88F6828**—Supports DDR3/DDR3L-1600/1866 and DDR4-1800
- **88F6811/88F6821**—Supports DDR3/DDR3L-1333
- **88F6W21**—Supports DDR3/DDR3L-1200
- Supports synchronous clock ratio of 1:N and 2:N between the CPU cores and the DRAM interface
- Supports asynchronous clock frequencies between the CPU cores and the DRAM interface
- Supports 1.5 and 1.35V for DDR3 (SSTL) and 1.2V for DDR4 (POD[1])
- Smart scheduler for transaction re-ordering and QoS
- Auto calibration of I/O output impedance
- Up to 8 Gb for DDR3 and 16 Gb for DDR4 (supports all DDR devices densities, according to the JEDEC standard)
- DDR3/DDR4 write and read leveling support
- Proprietary data and control eye pattern optimization process to improve signal integrity
- DDR3 address mirroring support
- Supports DDR3/DDR4 BL8
- Supports the 2T and 3T modes to enable high-frequency operation even under a heavy-load configuration
- Supports SDRAM bank interleaving
- 8/16 open pages per rank (DDR3 and DDR4, respectively)
- Up to a 128-byte burst per single memory
- Supports x8 and x16 memory devices
- Supports the following DDR4 features:
  - DDR4 PDA (Per DRAM Accessibility)
  - DDR4 MRS readout
  - DDR4 different bank group timings
  - Supports different pin mapping for DDR3/DDR4 optimization of board routing

1. Pseudo Open Drain

Copyright © 2015 Marvell

Gigabit Ethernet (GbE) Ports

- **88F6810**
  - 2 GbE ports (0 and 1)
  - SGMII / RGMII / MII interface
- **88F6820/88F6828**
  - 3 GbE ports (0–2)
  - SGMII / QSGMII / RGMII / MII interface
- **88F6811/88F6821/88F6W21**
  - 2 GbE ports (0 and 1)
  - SGMII / RGMII / MII interface
- SGMII supports 10/100/1000/2500 Mbps
- Full-wire speed receive and transmit of short packets
- Support for IEEE 1588v2 (PTP)
- DA filtering
- Hash-based statistical load balancing (RSS)
- Strict priority/WRR arbitration between 8 transmit queues with rate limiting
- Per queue egress rate shaping
- Support for queuing based on Marvell® DSA tag
- Support for jumbo frames (up to 10K) on both receive and transmit
- TCP/IP checksum acceleration
- Support for IEEE 802.3az (Energy-Efficient-Ethernet)
- Support for Wake-On-LAN

Hardware Buffer Management

- 4 independent buffer pools
- Up to 64 KB elements per pool
- Dedicated DMA engines for read-ahead or write-back from/to external memory

Copyright © 2015 Marvell

PCI Express (PCIe) Interface

- **88F6810/88F6811/88F6821**—Supports 3 ports x1 lanes
- **88F6W21**—Supports 2 ports x1 lanes
- **88F6820/88F6828**—Supports 1 port x4 lanes and additional 2 ports x1 lane or up to 4 ports x1 lane
- PCIe Gen 1.1 at 2.5 Gbps / Gen 2.0 at 5 Gbps signaling
- Supports Root Complex and Endpoint modes
- Lane polarity inversion/reversal support
- Maximum payload size of 128 bytes
- Single Virtual Channel (VC-0)
- Replay buffer support
- Extended PCIe configuration space
- Power management: L0s and L1 ASPM active power state support; software L1 and L2 support
- MSI / MSI-x support
- Error message support
- PCIe master specific features:
  - Host to PCIe bridge—translates CPU cycles to the PCIe memory or configuration cycles
  - Supports DMA bursts between memory and PCIe
  - Supports up to 4 outstanding read transactions
  - Maximum read request of up to 128 bytes
- PCIe target specific features:
  - Supports reception of up to 8 read requests
  - Maximum read request of up to 4 KB
  - Supports PCIe access to the device internal and external space

High-Speed Integrated SERDES Lanes

- **88F6810**—Integrates 5 low-power, high-speed SERDES PHYs, based on proven Marvell SERDES technology
- **88F6811/88F6821/88F6W21**—Integrates 4 low-power, high-speed SERDES PHYs, based on proven Marvell SERDES technology
- **88F6820/88F6828**—Integrates 6 low-power, high-speed SERDES PHYs, based on proven Marvell SERDES technology
- Diverse multiplexing options of PCIe, SATA 3.0, SGMII, QSGMII, and USB 3.0 interfaces

Copyright © 2015 Marvell
Document Classification: Proprietary Information

USB Controllers

- **88F6810/88F6820/88F6828**—3 USB ports configurable from the following 4 USB controllers:
  - 2 USB 3.0/2.0 Host compliant ports with integrated PHYs
  - 1 USB 3.0 Device compliant port with integrated PHYs
  - 1 USB 2.0 Host or Device compliant port with integrated PHY
- **88F6811/88F6821**—2 USB ports configurable from the following 3 USB controllers:
  - 2 USB 3.0/2.0 Host compliant ports with integrated PHYs
  - **88F6811/88F6821**—1 USB 3.0 Device compliant port with integrated PHYs
- As a Host:
  - Extensible Host Controller interface (xHCI) compatible
  - Supports direct connection to all device types (SS, HS, FS, LS)
  - Supports up to 64/16 independent endpoints (USB 3.0/USB 2.0, respectively)
- As a Device:
  - Connects to all host types (SS, HS, FS) and hubs
  - Supports up to 16/4 independent endpoints (USB 3.0/USB 2.0, respectively)
- Supports control, interrupt, bulk, and isochronous data transfers
- USB Power Management states:
  - U0 (Active), U1 (Idle), U2 (Deep Idle), and U3 (Suspended) for USB 3.0
  - Suspend and Resume for USB 2.0

2 Marvell® 6 Gbps (Gen 3x) SATA 3.0 Ports

- **88F6810/88F6820**—2 fully compliant SATA 3.0 ports
- **88F6828**—4 fully compliant SATA 3.0 ports
- **88F6811/88F6821**—2 fully compliant SATA 3.0 ports
- **88F6W21**—1 fully compliant SATA 3.0 port
- Supports communications speeds of 6.0 Gbps, 3.0 Gbps, and 1.5 Gbps
- Supports Gen 1x, Gen 2x, Gen 3
- Supports programmable transmitter signal levels
- Supports Native Command Queuing (NCQ) and First Party DMA (FPDAM) with up to 32 outstanding commands per port
- Supports AHCI 1.0 and IDE programming interfaces
- Supports Port Multiplier (PM)—Performs FIS-Based-Switching as defined in the SATA working group PM definition
- Port Selector (PS)—Issues the protocol-based Out-of-Band (OOB) sequence to select the active host port
- Supports external SATA (eSATA)
- Supports Partial and Slumber power management states
- Advanced interrupt coalescing
- Advanced drive diagnostics via the ATA SMART command

Copyright © 2015 Marvell

2 Cryptographic Engines

- Hardware implementation on Encryption/Decryption and Authentication engines to boost IP security (IPsec) and file encryption/decryption
- Dedicated DMA to feed the hardware engine with data from the internal SRAM memory or from the DDR memory
- Implements AES, DES, and 3DES encryption algorithms
- Implements SHA2, SHA1, and MD5 authentication algorithms

4 XOR DMA Engines

- RAID5 P (XOR parity) generate and repair up to 8 source blocks
- RAID6 P and Q generate and repair up to 8 source blocks
- Memory copy (DMA) acceleration
- iSCSI CRC-32 calculation
- Memory initialization function

TDM Controller

- Supports 2 independent VoIP channels
- Generic interface to standard SLIC / SLAC / DAA / codec devices
- Compatible with standard PCM highway formats
- Supports various bit clock rates (256 kHz to 8.192 MHz, in increments of powers of 2)

I^2S Input/Output and S/PDIF Output Audio Interfaces

- Sample rates of: 44.1, 48, and 96 kHz
- Compliant with IEC 60958-1, 60958-3, and 61937 specifications
- Supports plain I^2S, right-justified, and left-justified formats

Device Bus Controller

- 8/16-bit multiplexed address / data bus
- Supports different types of standard memory devices, such as NOR Flash and ROM
- Up to 5 chip selects with programmable timing

NAND Flash Controller

- Native support for NAND Flash with up to:
  - 8 KB page size
  - 16-bit ECC per page
- 4 chip selects with programmable timing and optional external wait-state control

2 SPI Ports

- General purpose 1-bit SPI interface
- SPI port has up to 4 chip selects

SD / SDIO / MMC / eMMC Host Interface

- 1-bit / 4-bit SDmem, SDIO 3.0
- 1-bit / 4-bit / 8-bit MMC 4.4 for eMMC
- SDR-50: Up to 100 MHz
- HS-DDR: Up to 50 MHz
- Hardware generate/check CRC on all command and data transactions on the card bus

Document Classification: Proprietary Information
Copyright © 2015 Marvell

2 UART Interfaces

- 16750 UART compatible
- Each port has:
  - 2 pins for transmit and receive operations
  - 2 pins for modem control functions

Advanced Power Management

- CPU Power Management states:
  - Deep Idle (CPU power down)
  - Idle (CPU clock gating)
  - Throttle (CPU reduced speed)
  - Run (CPU full speed)
- Standby with Real Time Clock (RTC) alarm or external wake-up options
- Wake-On-LAN (WOL) with lossless packet reception
- SDRAM Self Refresh and Power Down modes
- Selectable clock gating of different interfaces
- PCIe, SGMII, USB, and SATA SERDES shutdown
- Energy Efficient Ethernet (EEE)
- Various wake-up options

2 I^2C Interfaces

- General purpose I^2C master/slave ports
- EEPROM Serial initialization support

Real Time Clock

- External system wake-up
- Battery backup

Integrated BootROM

- Boot from Flash (SPI, parallel NOR, parallel NAND, serial NAND (SPI-NAND), and SDIO / MMC / eMMC)
- Boot from PCIe or SATA
- Secured boot flow

2 Integrated Low Drop Out (LDO) Controller

- On-chip LDO controller for external Positive-Negative-Positive (PNP) transistors
- Programmable output voltages of 1.8/2.5V with 50 mV resolution.

Multi-Purpose Pins (MPP)

- **88F6810/88F6820/88F6828**—60 MPP pins dedicated for peripheral functions and General Purpose I/O (GPIO)
- **88F6811/88F6W21/88F6821**—48 MPP pins dedicated for peripheral functions and General Purpose I/O (GPIO)
- Each pin independently configurable
- GPIO inputs configurable to register interrupts from external devices and generate maskable interrupts
- GPIO outputs configurable for PWM control of LED or fan operation

Copyright © 2015 Marvell

Clock Generation	■ Internal generation of CPU clock, Core clock, SDRAM clock, RGMII/MII clock, SERDES clock, TDM clock, Audio clock, and external Reference clock from a single 25-MHz reference clock ■ Supports internal generation of spread spectrum clocking on the CPU subsystem and SDRAM interface
Interrupts	ARM compliant Generic Interrupt Controller (GIC), version 1
Thermal Sensor	On-die Thermal Sensor with over-heat interrupt
Timers/Counters and Watchdog Timers	Integrated programmable 32/64-bit timers/counters and watchdog timers
Package	**88F6810/88F6820/88F6828**—TFBGA 17x17 mm package, 0.8 mm ball pitch **88F6811/88F6821/88F6W21**—TFBGA 14x14 mm package, 0.65 mm ball pitch

2.1 Device Differences

Table 1 lists the differences between the devices.

Table 1: Device Differences

	88F6810	88F6820	88F6828	88F6811	88F6821	88F6W21
JTAG Device ID	0x6820					
CPU						
Core Counts	Single	Dual	Dual	Single	Dual	Dual
CPU Architecture	Cortex A9	Cortex A9	Cortex A9	Cortex A9	Cortex A9	Cortex A9
L1 I/D cache	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB
Max Core Speed (GHz)	1.866 GHz	1.866 GHz	1.866 GHz	1.33 GHz	1.33 GHz	1.2 GHz
L2 Cache Size	1 MB	1 MB	1 MB	1MB	1MB	1MB
Ethernet						
GbE Ports	2	3	3	2	2	2
RGMII	2	2	2	1	1	1
SGMII	2	3	3	2	2	1
QSGMII	No	Yes	Yes	No	No	No
Buffer Management	Yes	Yes	Yes	Yes	Yes	Yes

Document Classification: Proprietary Information

Table 1: Device Differences (Continued)

	88F6810	88F6820	88F6828	88F6811	88F6821	88F6W21
SMI	1	1	1	1	1	1
Memory						
DDR3/3L/4	DDR3/3L/4 16/32-bit + ECC 800/933 MHz	DDR3/3L/4 16/32-bit + ECC 800/933 MHz	DDR3/3L/4 16/32-bit + ECC 800/933 MHz	DDR3/3L 16/32-bit + ECC 667 MHz	DDR3/3L 16/32-bit + ECC 667 MHz	DDR3/3L 16-bit + ECC 600 MHz
NAND	Yes	Yes	Yes	Yes	Yes	Yes
SPI	2	2	2	2	2	2
Device Bus	8/16-bit	8/16-bit	8/16-bit	8/16-bit	8/16-bit	8/16-bit
Co-processor						
Cryptographic Engine/ Security Accelerator	2	2	2	2	2	2
DMA/RAID5/ RAID6	4-Channel	4-Channel	4-Channel	4-Channel	4-Channel	4-Channel
Connectivity						
PCIe2.0	3 Ports x1	1 Port x4 + 2 Ports x1 or 4 Ports x1	1 Port x4 + 2 Ports x1 or 4 Ports x1	3 Ports x1	3 Ports x1	2 Ports x1
USB2.0 (Host/Device)	1	1	1	No	No	No
USB3.0 (Host)	2	2	2	2	2	2
USB3.0 (Device)	1	1	1	1	1	No
SATA3	2	2	4	2	2	1
SDIO3	Yes	Yes	Yes	Yes	Yes	Yes
I^2S / S/PDIF	Yes	Yes	Yes	No	No	No
2 Channel TDM	Yes	Yes	Yes	No	No	No
High Speed SERDES	5	6	6	4	4	4
Peripherals						
PTP	1	1	1	1	1	1
UART	2	2	2	2	2	2
I^2C	2	2	2	2	2	2

Copyright © 2015 Marvell

Table 1: Device Differences (Continued)

	88F6810	88F6820	88F6828	88F6811	88F6821	88F6W21
System Functions						
GPIO	60	60	60	48	48	48
RTC	Yes	Yes	Yes	Yes	Yes	Yes
AVS	Yes	Yes	Yes	Yes	Yes	Yes
eFuse	Yes	Yes	Yes	Yes	Yes	Yes
LDO	2	2	2	2	2	2
BootROM	Yes	Yes	Yes	Yes	Yes	Yes
PCIe Clock Outputs	2	2	2	1	1	1
Secure Boot	Yes	Yes	Yes	Yes	Yes	Yes
Technology						
Package	17x17 TFBGA, 0.8 mm	17x17 TFBGA, 0.8 mm	17x17 TFBGA, 0.8 mm	14x14 TFBGA, 0.65 mm	14x14 TFBGA, 0.65 mm	14x14 TFBGA, 0.65 mm
Process	28 LP	28 LP	28 LP	28 LP	28 LP	28 LP
I-Temp	Yes	Yes	Yes	Yes	Yes	Yes

Document Classification: Proprietary Information

3 Typical Applications and System Configurations

The devices are designed for a variety of Home and Enterprise applications. This section provides examples of such applications.

- Small/Medium Business (SMB) NAS
- Enterprise Wireless Access Point (WAP)
- Smart Gateway
- Network Video Recorder (NVR)
- Main CPU in a Control Plane

3.1 Small/Medium Business (SMB) NAS

The devices bring a combination of full SoC integration, superior performance, and a low total system cost for the Office/SMB 4-8 Bay NAS.

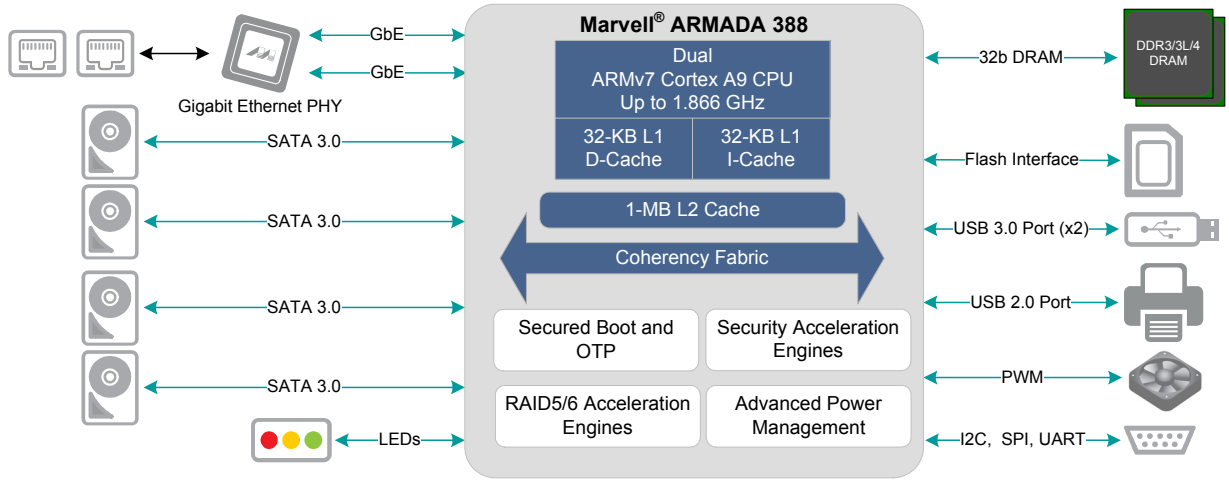
The Office NAS requires high reliability and high performance to support a large number of users in a secure environment.

The devices have all the relevant features, including SDRAM ECC protection, 2 Gbps AES engine, RAID5/6 acceleration (for fast RAID backup and repair), de-duplication hashing, secured boot and high NAS performance. The system cost is driven down with a 4-layer, mini-ITX form factor PCB, CPU fan-less operation, and with a very high level of BOM saving integrated functions.

The low active power consumption and the ultra-low standby power contribute significantly to meeting the energy standards requirements.

Figure 3 shows the 88F6828 (ARMADA® 388) in an SMB NAS application.

Figure 3: SMB NAS Application



3.2 Enterprise Wireless Access Point (WAP)

The devices are designed for meeting the requirements of a dual-band 802.11ac/n SMB/Enterprise class WAP application. It includes the relevant network and security acceleration hardware to perform up to 2Gbps line rate classified bridging between the wireless subsystem and the wired Ethernet links.

The device high performance CPU subsystem controls the packet flow within the SoC and prepares the descriptor queue for the Ethernet/Wireless MAC and intermediate acceleration engines. The integrated packet modification engine in Ethernet port 0 handles most of the data plane traffic, resulting in a significant reduction in CPU processing power. The devices support end-to-end Quality of Service (QoS), Link Aggregation (LAG), and guarantee prioritization between low-latency and best effort traffic flows.

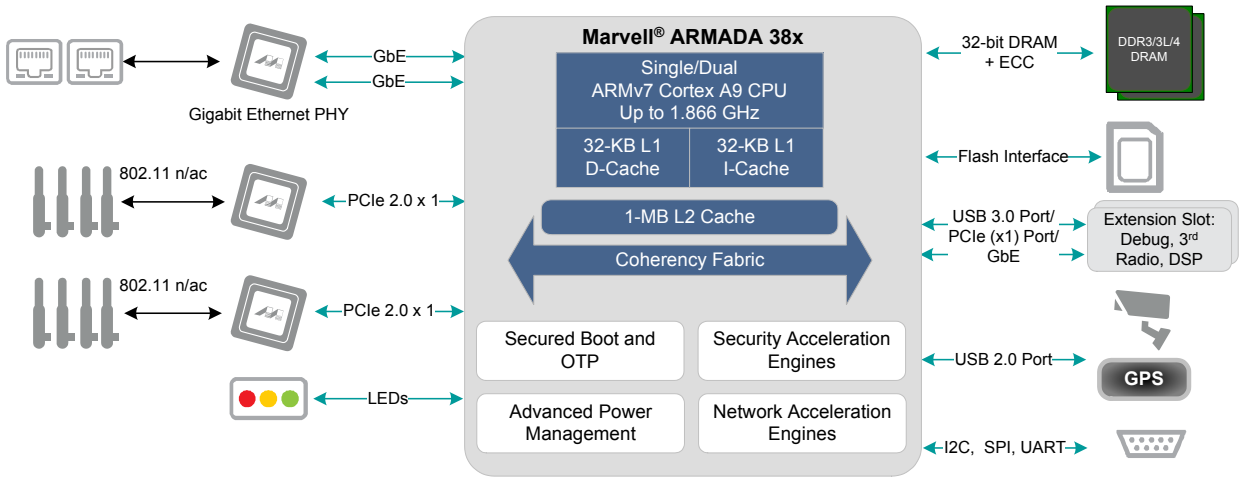
For RBOM saving, the devices support both DDR3/3L and DDR4 and a variety of Flash interfaces, providing system designers with the flexibility in cost and performance optimization.

The devices' low-power manufacturing technology combined with an enhanced power management architecture is the key for designing Power-over-Ethernet (PoE) capable systems. The devices deliver the best Performance per Watt solution for their class.

The devices, paired with the Marvell industry-leading Wireless and Wired networking solutions, deliver the right balance between performance, power, and price for indoor and outdoor 5/2.4G systems.

Figure 4 shows the devices in a Wi-Fi Access Point application.

Figure 4: Wi-Fi Access Point Application



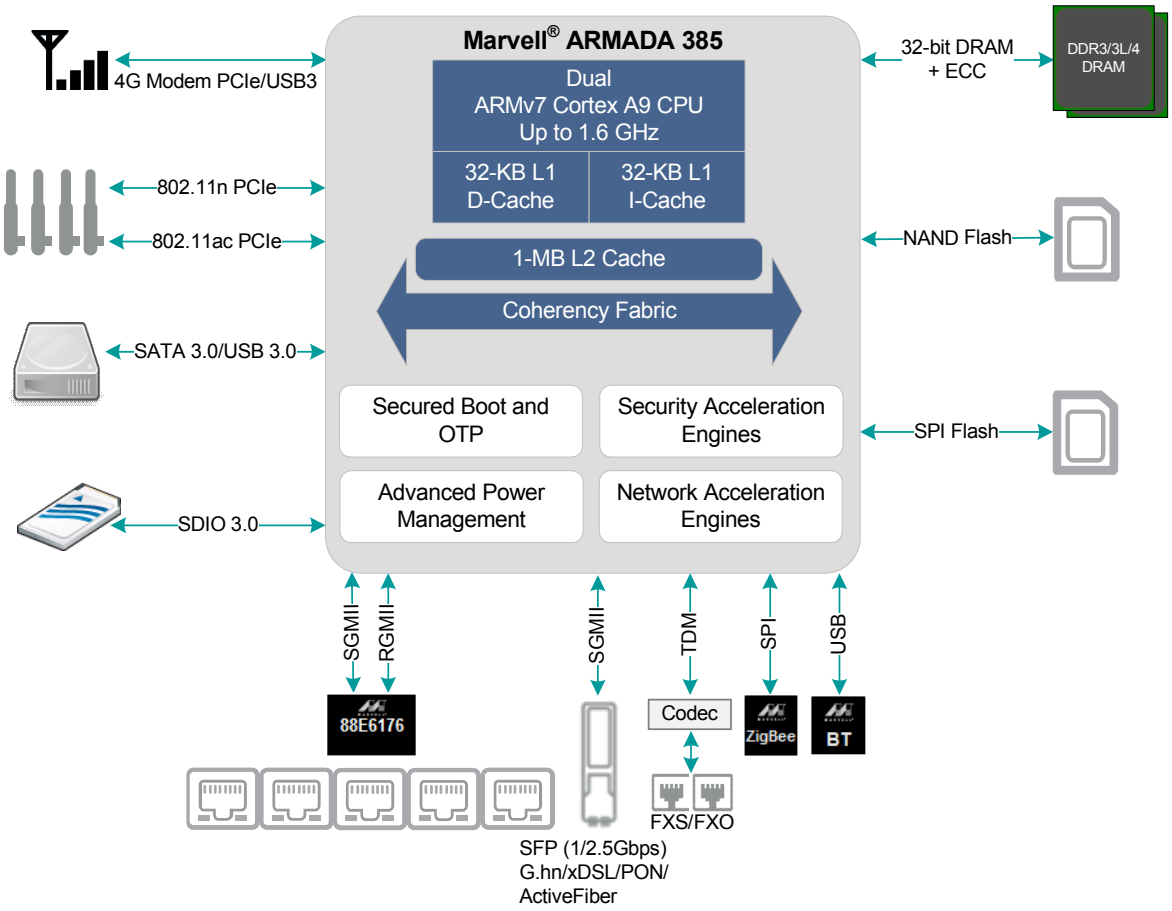
Copyright © 2015 Marvell

3.3 Smart Gateway

The devices are designed to power the IoT Smart Home and Smart Gateway applications. The ARMADA® 38x Family supports a wide set of acceleration and connectivity hardware enabling up to 2.5Gb/s indoor connectivity over the wireless (for example, Wi-Fi or LTE) and/or wireline infrastructure. The high performance cores enable a new level of computation-intensive networking applications, such as parental control and content sharing/synchronization.

Figure 5 shows the 88F6820 (ARMADA® 385) in a Smart Gateway application.

Figure 5: Smart Gateway Application

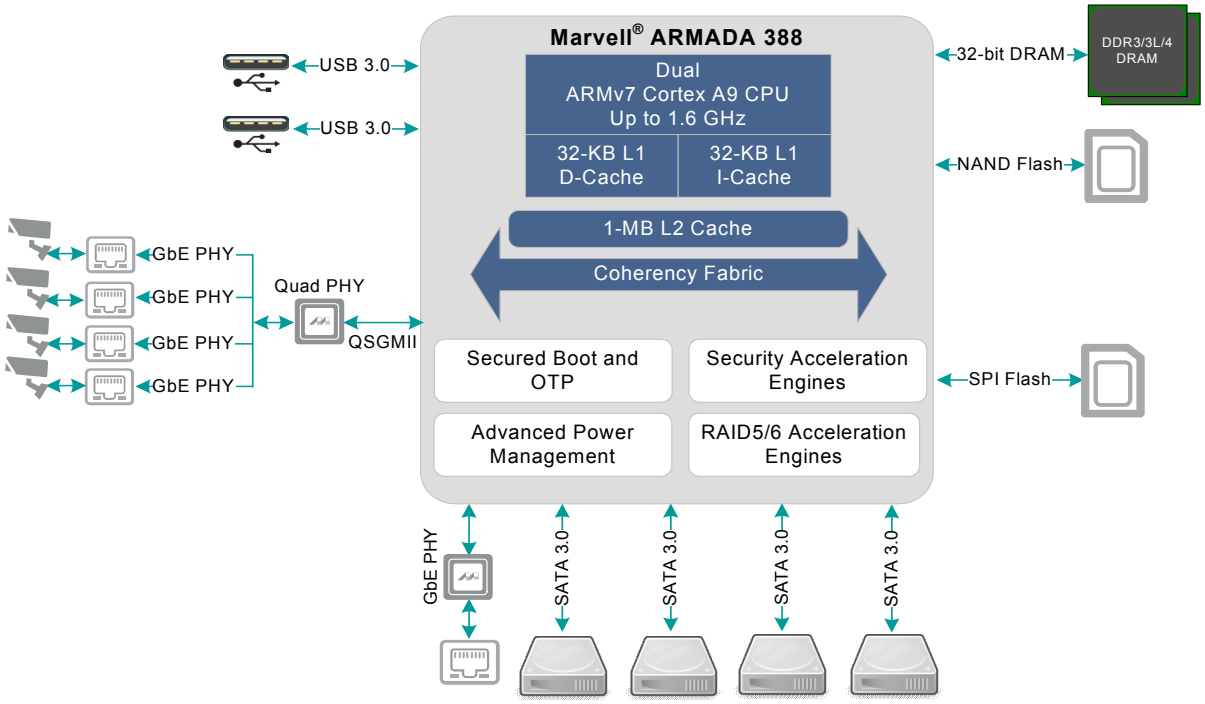


3.4 Network Video Recorder (NVR)

The devices support the Consumer and Small Medium Business (SMB) Network Video Recorder (NVR) applications. The device high-performance cores, optimized hardware for data storage, and high-performance networking unit enable highly efficient and cost-optimized NVR applications. The CPU cores perform real-time processing of the traffic ingressing via the integrated network ports and manage the data storage in the background, via the integrated SATA ports.

Figure 6 shows the 88F6828 (ARMADA® 388) in an NVR application.

Figure 6: NVR Application



Copyright © 2015 Marvell

3.5 Main CPU in a Control Plane

The devices, coupled with a high-performance CPU subsystem, offer an ideal choice for networking control planes, switch management, and line card management applications.

With tightly coupled, low-latency, high-bandwidth DRAM and ECC option, the high-speed CPU allows the devices to match the high-demand computing power and the reliability requirements of a control plane application. The scalability of single core and dual core makes it possible to operate the cores in SMP mode, and gain higher computing power than in a single CPU mode.

The devices can be used as control-data plane CPUs for low-end Access Routers (see Figure 7). In this case, the CPU is connected to the WAN modules via an external switch for data/control communication, and the PCIe is used for management, configuration, and data.

The third GE port can be used as a management port.

The integrated USB controller and PHY enable a convenient interface for software code upgrades, while the diverse Flash interfaces (NOR, NAND, SDIO/eMMC, or SPI) provide the system designer with the ability to choose the best boot option to fit the application needs and minimize the BOM cost.

Figure 7 shows the devices in a control plane CPU application.

Figure 7: Main CPU in a Control Plane Application

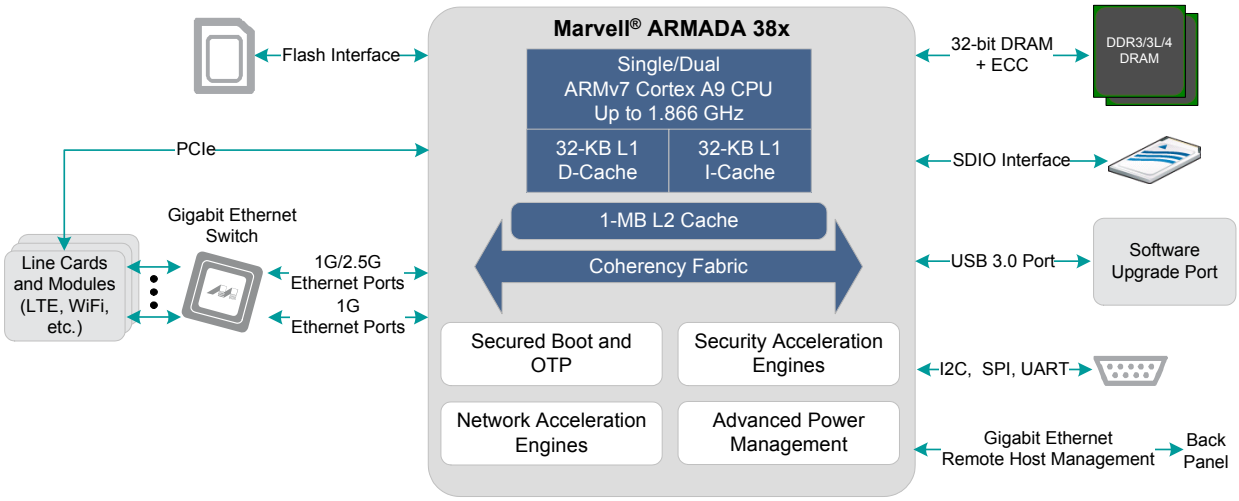
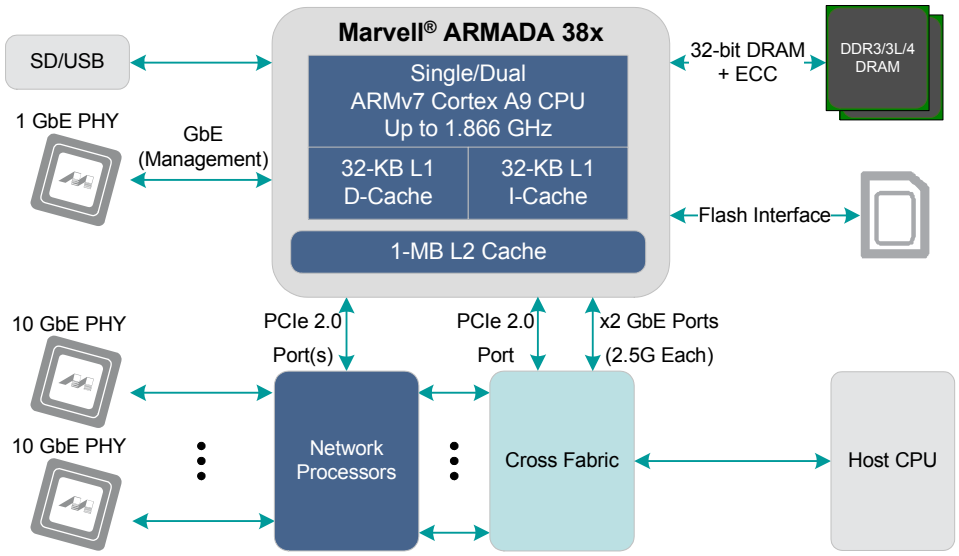


Figure 8 shows the devices a line card management application.

Figure 8: Line Card Management Application



Copyright © 2015 Marvell

4 Address Map

The devices have a fully programmable address map. There is a separate address map for each of the master units (units that can initiate transactions over the device Mbus). Each of the Mbus masters described below includes programmable address windows that enables it to access the different device resources.

The Mbus masters are:

- Cortex™-A9 Processor(s)
- PCI Express ports
- USB 2.0/3.0 MAC(s)
- SATA MAC ports
- GbE MAC ports
- Security Accelerator(s)
- Buffer Manager
- XOR DMA channels 0–3
- SDIO
- TDM
- Audio Unit (I^2S / S/PDIF)
- I^2C Serial Initialization
- General Management



Note

Throughout this section, the term BAR means Base Address Register.

4.1 Internal Address Map

The devices have a 1 MB internal register address space. The location of the internal memory space in the 4 GB address map is configured through a dedicated CPU address decoding window.

Table 2 lists the base address offset for each unit.

Table 2: Internal Address Map

Unit	Base Address	Maximum Address
DRAM	0x00000	0x07FFF
L2 Cache	0x08000	0x0BFFF
Generic Interrupt Controller (GIC)	0x0C100	0x0C1FF
Interrupt Distributor	0x0D000	0x0DFFF
Snoop Control Unit (SCU)	0x0F000	0x0F0FF
Shared Timers	0x0F200	0x0F2FF
Private Timers and Watchdog	0x0F600	0x0F6FF
Device Bus	0x10400	0x105FF
SPI Ports	0x10600	0x109FF

Copyright © 2015 Marvell

Table 2: Internal Address Map (Continued)

Unit	Base Address	Maximum Address
I2C Ports	0x11000	0x11FFF
UART Ports	0x12000	0x18FFF
GPIO, MPP, Miscellaneous	0x18000	0x1FFFF
CPU Subsystem	0x20000	0x2FFFF
Ethernet MAC Port 1	0x30000	0x33FFF
Ethernet MAC Port 2	0x34000	0x37FFF
PCI Express Port 1	0x40000	0x43FFF
PCI Express Port 2	0x44000	0x47FFF
PCI Express Port 3	0x48000	0x4BFFF
USB 3.0 Device port 0	0x50000	0x57FFF
USB 2.0 Host/Device port 0	0x58000	0x5FFFF
XOR Engines	0x60000	0x6FFFF
Ethernet MAC Port 0	0x70000	0x7FFFF
PCI Express Port 0	0x80000	0x8FFFF
Cryptographic Engine Security Accelerator (CESA)	0x90000	0x9FFFF
Common PHY Configuration	0xA0000	0xAFFFF
SATA Port 0	0xA3800	0xA317F
SATA Port 1	0xA3800	0xA3BFF
TDM	0xB0000	0xBFFFF
UTMI Configuration	0xC0000	0xC7FFF
Buffer Manager	0xC8000	0xCFFFF
NAND Flash Controller	0xD0000	0xD7FFF
SDIO	0xD8000	0xDFFFF
SATA port 2	0xE0100	0xE017F
SATA port 3	0xE0180	0xE01FF
DFX Server	0xE4000	0xE7FFF
eFuse	0xE0000	0xE3FFF
Audio (S/PDIF / I2S)	0xE8000	0xEBFFF
USB 3.0 Host port 0	0xF0000	0xF7FFF
USB 3.0 Host Port 1	0xF8000	0xFFFFF

Copyright © 2015 Marvell

4.2 Cortex™-A9 Processor Address Decoding

The Cortex™-A9 Processor address decoding is performed in 2 steps:

1. Decode the primary target region (DRAM or other device units) according to:
   - SoC Window Control Register (Table 161 p. 465)
   - SoC Window Base Register (Table 162 p. 466)
2. Decode the secondary region (DRAM Chip Select or SoC device units) using 23 address windows as follows:
   - 2 windows dedicated for core access to the CSn DRAM chip selects: Marvell® recommends using DRAM Windows 0 and 1. Before implementing a design using DRAM Windows 2 and 3 contact a Marvell field applications engineer.
   - 1 window for core access to the device internal registers space
   - 20 configurable windows for core access to the rest of the device's resources



Note

The order of the address decoding from the original address to the final destination is accomplished in this order:

1. Internal register
2. DRAM
3. Other device resources (I/O mapping)

DRAM Windows

The device supports up to 4 GB of DRAM memory space.

Each DRAM window can have an address space of between 16 MB and 4 GB. Each window is defined by a Base and Size registers. The base and size are 8-bits wide, corresponding to address bits[31:24]. The size must be programmed as a set of 1s (starting from the LSB) followed by a set of 0s. The set of 1s defines the size. For example, if Size[7:0] is set to 0x0F, it defines a size of 256 MB (number of 1s is 4, 2^4 x 16 MB = 256 MB).

By default, each of the DRAM windows corresponds to a different DRAM chip select (M_CSn[1:0]). However, each window can be set to support any of the DRAM chip selects. This feature provides flexibility in mapping DRAM space.



Note

The CPU can also be configured to access the DRAM through the Mbus as any other interface in the device, instead of through the CSn dedicated DRAM windows.

Do not overlap the CPU DRAM dedicated windows with CPU windows that access DRAM through the Mbus.

Address Windows

Apart from the DRAM windows, each of the configurable address windows can have a minimum of 1 MB of address space and up to 4 GB. Each window is defined by a Window Base register and by the <Size> field in the Window Control register. The Base and Size are 16 bits wide, corresponding to address bits[31:16]. The Size must be programmed as a set of 1s (starting from the LSB) followed by a set of 0s. The set of 1s defines the size. For example, if Size[15:0] is set to 0x03FF, it defines a size of 64 MB (number of 1s is 10, 2^{10} x 64 KB = 64 MB).

The device's internal registers window has a 1 MB fixed size (It includes a Base register only, but no Size register).

Copyright © 2015 Marvell

Note

Only part of this space is populated with registers.

Address Decoding

Address decoding starts with the address being compared with the values in the various Base Address registers. The Size sets the address bits that are significant for the comparison. In the previous example of a 64 MB size, the CPU address bits[31:26] are compared with the Base Address bits[15:10] (the Size masks address bits[25:0]). An address is considered as a window hit if it matches the Base Address register bits (the bits that are not masked by the Size).

Note

- Never program the Base and Size registers so that they result in an address windows overlap.
- The address decoding scheme restricts the address window to a size of 2^n and to a start address aligned to the window size.

Upon a hit in one of the 20 configurable windows, the transaction is forwarded to a specific target interface specified by the Window Control register's <Target> bits[7:4], and with specific transaction attributes specified by the Window Control register's configurable <Attr> bits [15:8].

Table 3 is a summary of target units IDs and attributes. It is relevant for access from the CPU core only.

Table 3: CPU Interface Mbus Decoding Units IDs and Attributes

Field	Description
Target Unit ID	0x0 = DRAM 0x1 = Device bus, bootROM, SPI, UART, GPIO, MPP, and Miscellaneous 0x2 = CPU-IO Bridge 0x3 = Gigabit Ethernet MAC ports 1 and 2 0x4 = PCI Express ports 1, 2, and 3 0x5 = USB 3.0 Device and USB 2.0 MACs 0x6 = XOR 0 and 1 0x7 = Gigabit Ethernet MAC port 0 0x8 = PCI Express port 0 0x9 = Cryptographic Engine Security Accelerators 0 and 1 0xA = SATA 3 ports 0 and 1 0xB = TDM controller 0xC = Buffer Manager 0xD = NAND Flash controller and SDIO 3 controller 0xE = SATA 3 ports 2 and 3 (88F6828 only), Audio unit (S/PDIF / I^2S) 0xF = USB 3.0 Host ports 0 and 1

Copyright © 2015 Marvell

Table 3: CPU Interface Mbus Decoding Units IDs and Attributes (Continued)

Field	Description
Attributes[7:0]	**If the target is the DRAM interface:** Bits[7:0] = Memory chip select: 0x0E = M_CS[0] request in DRAM controller 0x0D = M_CS[1] request in DRAM controller All other values are reserved. **If the target is the Device bus interface:** Bits[7:0] = chip select: 0x1 = Reserved 0x1E = SPI0 CS0 request 0x5E = SPI0 CS1 request 0x9E = SPI0 CS2 request 0xDE = SPI0 CS3 request 0x1A = SPI1 CS0 request 0x5A = SPI1 CS1 request 0x9A = SPI1 CS2 request 0xDA = SPI1 CS3 request 0x3E = DevCS[0] request 0x3D = DevCS[1] request 0x3B = DevCS[2] request 0x37 = DevCS[3] request 0x2F = BootCS request 0x1D = bootROM request All other values are reserved. **NOTE:** When addressing SPI and working in Single Window mode, configure these bits to (Section 21.5.3, SPI Timing, on page 271): 0x1E for addressing SPI0 CS 0–3 0x1A for addressing SPI1 CS 0–3 When addressing SPI and working in Double Window mode configure these bits to: 0x1E for addressing SPI0 CS0–3 0x1A for addressing SPI1 CS0–3

Table 3: CPU Interface Mbus Decoding Units IDs and Attributes (Continued)

Field	Description
Attributes[7:0] (Continued)	**If the target is PCI Express interface:** Bits[2:0] = Reserved. Must be 0x0. Bit[3] = Memory/IO select: 0x0 = I/O 0x1 = Memory Bits[7:4] = PCIE port select (Any other value is reserved): 0xE = PCIe port X **NOTE:** • When the "Target Unit ID" is 0x8 (PCIe0), bits[7:4] should be 0xE. • When the "Target Unit ID" is 0x4 (PCIe1/2/3), bits[7:4] should be: 0xE = PCIe port1 0xd = PCIe port2 0xb = PCIe port3 **If the target is the Security Accelerator SRAM:** Bits[1:0] = Data swapping 0x0 = Byte swap 0x1 = No swap 0x2= Byte and word swap 0x3 = Word swap Bit[4:2] = CESA select: 0x6 = CESA0 0x5 = CESA1 Bits[7:5] = Reserved. Must be 0. **If accessing any other target unit set bits[7:0] to 0x00.**

There are 23 address windows:

- 1 internal window
- 2 DRAM windows
- 20 IO windows

Each of the CSn[1:0] DRAM windows and the 20 configurable windows has an Enable bit (Size register's <En> bit[0]). When set to 1, the window is enabled. When cleared to 0, the window is disabled and does not take part in the address decoding process. A CPU address that is going through the address decoding scheme is 32 bits wide.

Table 4 shows the default Cortex™-A9 Processor memory map following reset de-assertion.

Table 4: Cortex™-A9 Processor Default Address Map

Decoder	Address Range	Size	Target Unit ID	Target Attribute
DRAM M_CSn[0]	0x0 to 0x0.0FFF.FFFF **NOTE:** Disabled by default.	256 MB	NA	NA
DRAM M_CSn[1]	0x0.1000.0000 to 0x0.1FFF.FFFF **NOTE:** Disabled by default.	256 MB	NA	NA
Reserved	0x2000.0000 to 0x7FFF.FFFF		NA	NA
Configurable Window 0	0x8000.0000 to 0x87FF.FFFF	128 MB	0x4 (PCIe1/2/3)	0xE8 (Memory, port 1)

Copyright © 2015 Marvell

Table 4: Cortex™-A9 Processor Default Address Map (Continued)

Decoder	Address Range	Size	Target Unit ID	Target Attribute
Configurable Window 1	0x8800.0000 to 0x8FFF.FFFF	128 MB	0x4 (PCIe1/2/3)	0xE8 (Memory, port 1)
Configurable Window 2	0x9000.0000 to 0x97FF.FFFF	128 MB	0x4 (PCIe1/2/3)	0xE8 (Memory, port 1)
Configurable Window 3	0x9800.0000 to 0x9FFF.FFFF	128 MB	0x4 (PCIe1/2/3)	0xE8 (Memory, port 1)
Configurable Window 4	0xA000.0000 to 0xA7FF.FFFF	128 MB	0x8 (PCIe0)	0xE8 (Memory, port 0)
Configurable Window 5	**NOTE:** Disabled by default.		NA	NA
Configurable Window 6	**NOTE:** Disabled by default.		NA	NA
Configurable Window 7	**NOTE:** Disabled by default.		NA	NA
Reserved	0xC010.0000 to 0xC800.FFFF		NA	NA
Configurable Window 8	0xC801.0000 to 0xC801.FFFF	64KB	0x9 (Security Accelerator SRAM)	0x01 (Engine 0, no data swap)
Reserved	0xC802.0000 to 0xCFFF.FFFF		NA	NA
Internal Registers	0xD000.0000 to 0xD00F.FFFF	1 MB	NA	NA
Reserved	0xD033.0000 to 0xD37F.FFFF		NA	NA
Configurable Window 9	0xD800.0000 to 0xDFFF.FFFF	128 MB	0x1 (Device bus)	0x2F (BOOT_CS)
Configurable Window 10	0xE000.0000 to 0xE7FF.FFFF	128 MB	0x1 (Device bus)	0x3E (DEV_CS[0])
Configurable Window 11	0xE800.0000 to 0xEFFF.FFFF	128 MB	0x1 (Device bus)	0x3D (DEV_CS[1])
Configurable Window 12	0xF000.0000 to 0xF7FF.FFFF	128 MB	0x1 (Device bus)	0x3B (DEV_CS[2])
Configurable Window 13	0xF800.0000 to 0xFFFF.FFFF	128 MB	0x1 (Device bus)	According to boot device type: 0x2F (= DevBOOTCS) or 0x1E (= SPI0 ROM) or 0x1D (= bootROM)
Configurable Window 14	0xD400.0000 to 0xD7FF.FFFF	64 MB[1]	0x1 (Device bus)	0x1E (SPI0 CS[0])
Configurable Window 15	0xD030.0000 to 0xD030.FFFF **NOTE:** Disabled by default.	64 KB	NA	NA

Copyright © 2015 Marvell

Table 4: Cortex™-A9 Processor Default Address Map (Continued)

Decoder	Address Range	Size	Target Unit ID	Target Attribute
Configurable Window 16	0xD031.0000 to 0xD031.FFFF **NOTE:** Disabled by default.	64 KB	NA	NA
Configurable Window 17	0xD032.0000 to 0xD032.FFFF **NOTE:** Disabled by default.	64 KB	NA	NA
Configurable Window 18	0xD380.0000 to 0xD3FF.FFFF	8 MB	0xC (Buffer Manager)	0x00
Configurable Window 19	0x0000.0000 to 0x07FF.FFFF **NOTE:** Use for DRAM training. After the training completes, the user may disable this window and instead enable the DRAM_M_CSn windows.	128 MB	0x0 (DRAM)	0x0E (M_CS[0])

1. In boot from 32-bit SPI, this window is disabled by bootROM firmware and replaced by Window 7.

The device internal registers space has a fixed size of 1 MB, even though only part of this space is actually populated by the device's internal registers. Upon a write to a non-implemented register, data is discarded. A read to a non-implemented register returns undefined data.

The registers are spread around the device's different units (distributed register file). Unless specified, ordering is not guaranteed on 2 write requests to registers of different units. If ordering is required on registers that are not specified to keep order, perform a read after each write.

Note

Access to internal registers is limited to WORD boundary (no support of burst access to register files). This means that register file space must never be cacheable.

4.2.1 Cortex™-A9 Processor-to-PCI Express Address Remapping

The device supports address remapping on Cortex™-A9 Processor access to a directly accessed interface (PCI Express, Serial and Parallel NOR Flash). The address remapping de-couples the memory map allocation of the core and the address space of the PCI Express or Flash memory devices. This enables relocating a CPU-to-PCI Express address window to a new location in the PCI Express address space.

Configurable windows 7–0 have associated remap registers. Upon a hit in one of these windows, the upper bits of the CPU address are replaced by the corresponding bits of the Remap Low register, before being transferred to the PCI Express interface unit. The number of bits to be replaced is determined by the Size register. For example, with a 64-MB window, the CPU address bits[31:26] are replaced with bits[31:26] of the Remap Low register.

Copyright © 2015 Marvell

Each of these windows also has a 32-bit Remap High register that can be used for 64-bit addressing on the PCI Express interface. When this register is not cleared to 0, a CPU address hit in this window results in the device's PCI Express master generating a 64-bit addressing transaction.



Note

Address window 13 has remap registers as well. They can be useful to allow boot from a 32b address SPI flash.

4.2.2 CPU Address Decoding Errors

In the case of an address decoding error (for example, no hit in any of the address windows):

1. An interrupt is set.
2. If it is a write transaction, it is discarded.
   If it is a read, dummy data is driven back to the Cortex™-A9 Processor with an erroneous data indication (resulting in CPU exception).

4.3 PCI Express Address Decoding

Each of the device's PCI Express ports has its own address map. The PCI Express interface address map consists of 3 Base Address Registers (BARs) that map the device address space. One BAR is dedicated to the device's internal registers space. The other 2 BARs are further sub-decoded by 6 programmable address windows to the different interfaces of the device.

The 3 BARs are 64-bit wide BARs. The internal registers space has a fixed size of 1 MB. The other 2 BARs have corresponding Size registers. Each window can have a minimum of 64 KB of address space and up to 4 GB of space.

PCI Express address decoding is similar to the CPU address decoding scheme. An address is considered as a window hit if it matches the Base Address register bits. These are the bits not masked by the Size register.



Note

- Do not program the Base and Size registers so that they result in an address window overlap.
- The PCI Express address decoding scheme restricts the address window to a size of 2^n, and to a start address that is aligned to the window size.

Each of the 2 BARs has an Enable bit. If a BAR is disabled, no address decoding is performed against it.

Upon an address hit, the address is further sub-decoded against 6 address windows. Each of these windows also consists of Base and Size registers. Based on this address decoding, the transaction is forwarded to a specific target interface (for example, the DRAM controller), with specific transaction attributes (for example, M_CS[0]).

Copyright © 2015 Marvell

Table 5 provides a summary of target unit IDs and attributes.

Table 5: Unit IDs and Attributes

Field	Description
Unit ID	0x0 = DRAM 0x1 = Device bus, bootROM, SPI, UART, GPIO, MPP, and Miscellaneous 0x2 = CPU-IO Bridge 0x3 = Gigabit Ethernet MAC ports 1 and 2 0x4 = PCI Express ports 1, 2, and 3 0x5 = USB 3.0 Device and USB 2.0 MACs 0x6 = XOR 0 and 1 0x7 = Gigabit Ethernet MAC port 0 0x8 = PCI Express port 0 0x9 = Cryptographic Engine Security Accelerators 0 and 1 0xA = SATA 3 ports 0 and 1 0xB = TDM controller 0xC = Buffer Manager 0xD = NAND Flash controller and SDIO3 controller 0xE = SATA 3 ports 2 and 3 (88F6828 only), Audio unit (S/PDIF / I^2S) 0xF = USB 3.0 Host ports 0 and 1

Copyright © 2015 Marvell

Table 5: Unit IDs and Attributes (Continued)

Field	Description
Attributes[7:0]	**If the target is the DRAM bus interface:** Bits[3:0] = Memory chip select: 0xE = M_CS[0] request in DRAM controller 0xD = M_CS[1] request in DRAM controller All other values are reserved. Bits[5:4] = Shared Memory Area with CPU (Coherency Status). Shared area forces sending the transaction through the coherency block. A Non-Shared transaction is driven directly to DRAM. 0x0 = Non-Shared transaction 0x1 = Shared transaction and no L2 read/write allocate (no L2 Deposit) 0x2 = Reserved 0x3 = Shared transaction and L2 write allocate (L2 Deposit) **If the target is the SPI interface:** Bits[7:0] = chip select: 0x1 = Reserved 0x1E = SPI0 CS0 request 0x5E = SPI0 CS1 request 0x9E = SPI0 CS2 request 0xDE = SPI0 CS3 request 0x1A = SPI1 CS0 request 0x5A = SPI1 CS1 request 0x9A = SPI1 CS2 request 0xDA = SPI1 CS3 request **If the target is the Device bus interface:** 0x3E = DevCS[0] request 0x3D = DevCS[1] request 0x3B = DevCS[2] request 0x37 = DevCS[3] request 0x2F = BootCS request 0x1D = bootROM request All other values are reserved.

Copyright © 2015 Marvell

Table 5: Unit IDs and Attributes (Continued)

Field	Description
Attributes[7:0] (Continued)	**If the target is PCI Express interface:** Bits[2:0] = Reserved. Must be 0x0. Bit[3] = Memory/IO select: 0x0 = I/O 0x1 = Memory Bits[7:4] = PCIe port select 0xE= PCIe port X **NOTE:** • When the "Target Unit ID" is 0x8 (PCIe0), bits[7:4] should be 0xE. • When the "Target Unit ID" is 0x4 (PCIe1/2/3), bits[7:4] should be: 0xE = PCIe port1 0xd = PCIe port2 0xb = PCIe port3 **If the target is the Security Accelerator SRAM:** Bits[1:0] = Data swapping 0x0 = Byte swap 0x1 = No swap 0x2 = Byte and word swap 0x3 = Word swap Bit[4:2] = CESA select: 0x6 = CESA0 0x5 = CESA1 Bits[7:5] = Reserved. Must be 0. **If accessing any other target unit, set bits[7:0] to 0x00.**

4.3.1 PCI Express to Memory Address Remapping

The device supports PCI Express address remapping. Each of the 6 PCI Express address windows has a Remap Register associated with it. The upper bits of the address, that are found to be a hit in one of the PCI windows, are replaced by the corresponding bits of the Remap register before being transferred to the target interface. The number of bits to be replaced is determined according to the Size register.

Windows 4 and 5 also have Remap High registers. This is useful for 64-bit addressing if the target interface supports 64-bit addressing (for example, PCI Express to PCI Express bridging).

Each Remap register has an Enable bit. If it is disabled, no remap action takes place and the original address is transferred to the destination.

Copyright © 2015 Marvell

4.3.2 PCI Express Address Decoding Errors

There are 2 situations where a PCIe decoding error may occur. This describes what occurs in each situation.

- If the device PCI Express port receives a transaction that does not match any of the BARs:
  1. An interrupt is set and the error is registered.
  2. The transaction is terminated as an unsupported request.
- If the device's PCI Express port receives a transaction that does hit one of the BARs, but does not match any of the sub-decoding windows:
  1. An interrupt is set and the error is registered.
  2. The transaction is forwarded to a default target, as defined in the PCI Express Default Window Control Register (Table 375 p. 581).



Note

Set the PCI Express Default Window Control Register (Table 375 p. 581) to point to a dummy target device, so no destructive operation is performed due to the address decoding error.

4.4 XOR DMA Address Decoding

The 2 XOR DMAs share a single address decoding logic consisting of 8 address windows. Whenever 1 of the ports generates a read or a write transaction (for example, fetch descriptor), the address is compared against these address windows, to determine which interface must be accessed.

The address decoding scheme is shown in Table 5.

4.5 USB Address Decoding

The address decoding logic for the USB port consists of the following address windows:

- USB2 MAC uses 4 address decoding windows.
- USB3-Host and Device use 8 address decoding windows.

Whenever 1 of the ports generates a read or a write transaction (for example, fetch descriptor), the address is compared against these address windows, to determine which interface must be accessed.

The address decoding Target ID and Attributes encoding is shown in Table 5.



Note

The USB port is restricted to access only the DRAM interface (M_CS[1:0]). Setting USB port address decoding windows differently results in unpredictable behavior.

Copyright © 2015 Marvell

4.6 SATA Address Decoding

The SATA port DMA uses an address decoding logic consisting of 4 address windows. The address decoding Target ID and Attributes encoding is shown in Table 5.

Note The SATA port is restricted to access only the DRAM interface (M_CS[1:0]). Setting SATA port address decoding windows differently results in unpredictable behavior.

4.7 Security Accelerator Address Decoding

The Security Accelerator DMA uses an address decoding logic consisting of 4 address windows.

The address decoding Target ID and Attributes encoding is shown in Table 5.

Note The Security Accelerator DMA is restricted to access only the DRAM interface (M_CS[1:0]). Setting the Security Accelerator DMA address decoding windows differently results in unpredictable behavior.

4.8 SDIO Address Decoding

The SDIO unit uses an address decoding logic consisting of 8 address windows.

The address decoding scheme is shown in Table 5.

Note The SDIO unit is restricted to access only the DRAM interface (M_CS[1:0]). Setting SDIO unit address decoding windows differently results in unpredictable behavior.

4.9 TDM Address Decoding

The TDM unit uses an address decoding logic consisting of 4 address windows.

The address decoding scheme is shown in Table 5.

Note The TDM unit is restricted to access only the DRAM interface (M_CS[1:0]). Setting TDM unit address decoding windows differently results in unpredictable behavior.

4.10 I^2C Address Decoding

The device's I^2C serial ROM initialization allows for access to the entire device's internal and external resources (see the Reset Pins and Configuration section in the device *Hardware Specifications*).

The serial ROM initialization access to the device resources consists of a 32-bit address followed by 32-bit data. Bit[0] of the transaction address defines whether it is a register access (bit cleared to 0) or an access to one of the external resources (bit set to 1). If it is an external resource, the exact interface target unit and attributes are defined via the <Attr> and <UnitID> fields in the I^2C Configuration Debug register.

Document Classification: Proprietary Information

By default, the unit ID is cleared to 0x0 (DRAM interface). If it is necessary to access the Device bus, for example, set the unit ID to 0x1. Once this is set, every external access is forwarded to the Device bus.

The Serial ROM initialization address decoder also includes a Remap High register. Use that register to generate an address beyond the standard 4 GB space. This is useful for 64-bit PCI Express addressing.

4.11 Audio Address Decoding

The Audio unit uses an address decoding logic consisting of 2 address windows.

The address decoding scheme is shown in Table 5.



Note

The Audio unit is restricted to access only the DRAM interface (M_CS[1:0]). Setting Audio unit address decoding windows differently results in unpredictable behavior.

5 System Considerations

This section describes the ARMADA® 38x Family system considerations, including data integrity and Big and Little Endian byte ordering.

5.1 Data Integrity

The device supports data integrity on most of its external interfaces, as follows:

- LCRC checking and generation on the PCI Express interface
- CRC checking and generation on the Ethernet, SATA, SDIO, and USB ports
- ECC checking and generation on the DRAM interface
- ECC checking and generation on the NAND interface

The device supports parity protection on CPU core L1 and L2 caches. In addition, the device supports parity protection on the internal data paths.

5.1.1 Cache Protection

The CPU L0, L1, and L2 cache supports Data and Tag RAM parity protection for both Instruction and Data caches. For further details, see the core processor specifications (see Related Documents on page 20).

Level 2 cache supports Tag RAM parity protection.

5.1.2 SDRAM ECC

The device implements Error Checking and Correction (ECC) on accesses to the SDRAM. It supports detection and correction of one data bit errors, and detection of 2 errors (SEC/DED).ECC is supported on 32 and 16 bits DRAM width options.

5.1.2.1 ECC Calculation

ECC is calculated on a 64-bit vector. When using 16- or 32-bit DRAM, the calculation is performed assuming that the unused bits are all zeros. Each of the 64 data bits and 8 check bits have a unique 8-bit ECC check code, as shown in Table 6. For example, data bit 12 has the check value of 01100001, and check bit 5 has the check value of 00100000.

Table 6: ECC Code Matrix

Check Bit	Data Bit	ECC Code Bits								Number of 1s in Syndrome
		7	6	5	4	3	2	1	0	
NOTE: For the 16b DRAM, the upper bits (data[63:16]) are zero in this device. For the 32b DRAM, the upper bits (data[63:32]) are zero in this device.										
	63	1	1	0	0	1	0	0	0	3
	62	1	1	0	0	0	1	0	0	3
	61	1	1	0	0	0	0	1	0	3
	60	1	1	0	0	0	0	0	1	3

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 6: ECC Code Matrix (Continued)

Check Bit	Data Bit	ECC Code Bits								Number of 1s in Syndrome
		7	6	5	4	3	2	1	0	
	59	1	1	1	1	0	1	0	0	5
	58	1	0	0	0	1	1	1	1	5
4		0	0	0	1	0	0	0	0	1
3		0	0	0	0	1	0	0	0	1
	57	1	1	1	0	0	0	0	0	3
	56	1	0	1	1	0	0	0	0	3
	55	0	0	0	0	1	1	1	0	3
	54	0	0	0	0	1	0	1	1	3
	53	1	1	1	1	0	0	1	0	5
	52	0	0	0	1	1	1	1	1	5
5		0	0	1	0	0	0	0	0	1
2		0	0	0	0	0	1	0	0	1
	51	1	0	0	0	0	1	1	0	3
	50	0	1	0	0	0	1	1	0	3
	49	0	0	1	0	0	1	1	0	3
	48	0	0	0	1	0	1	1	0	3
	47	0	0	1	1	1	0	0	0	3
	46	0	0	1	1	0	1	0	0	3
	45	0	0	1	1	0	0	1	0	3
	44	0	0	1	1	0	0	0	1	3
	43	1	0	1	0	1	0	0	0	3
	42	1	0	1	0	0	1	0	0	3
	41	1	0	1	0	0	0	1	0	3
	40	1	0	1	0	0	0	0	1	3
	39	1	0	0	1	1	0	0	0	3
	38	1	0	0	1	0	1	0	0	3
	37	1	0	0	1	0	0	1	0	3
	36	1	0	0	1	0	0	0	1	3

Copyright © 2015 Marvell

Table 6: ECC Code Matrix (Continued)

Check Bit	Data Bit	ECC Code Bits								Number of 1s in Syndrome
		7	6	5	4	3	2	1	0	
	35	0	1	0	1	1	0	0	0	3
	34	0	1	0	1	0	1	0	0	3
	33	0	1	0	1	0	0	1	0	3
	32	0	1	0	1	0	0	0	1	3
	31	1	0	0	0	1	0	1	0	3
	30	0	1	0	0	1	0	1	0	3
	29	0	0	1	0	1	0	1	0	3
	28	0	0	0	1	1	0	1	0	3
	27	1	0	0	0	1	0	0	1	3
	26	0	1	0	0	1	0	0	1	3
	25	0	0	1	0	1	0	0	1	3
	24	0	0	0	1	1	0	0	1	3
	23	1	0	0	0	0	1	0	1	3
	22	0	1	0	0	0	1	0	1	3
	21	0	0	1	0	0	1	0	1	3
	20	0	0	0	1	0	1	0	1	3
	19	1	0	0	0	1	1	0	0	3
	18	0	1	0	0	1	1	0	0	3
	17	0	0	1	0	1	1	0	0	3
	16	0	0	0	1	1	1	0	0	3
	15	0	1	1	0	1	0	0	0	3
	14	0	1	1	0	0	1	0	0	3
	13	0	1	1	0	0	0	1	0	3
	12	0	1	1	0	0	0	0	1	3
	11	1	1	1	1	1	0	0	0	5
	10	0	1	0	0	1	1	1	1	5
7		1	0	0	0	0	0	0	0	1
0		0	0	0	0	0	0	0	1	1

Document Classification: Proprietary Information

Table 6: ECC Code Matrix (Continued)

Check Bit	Data Bit	ECC Code Bits								Number of 1s in Syndrome
		7	6	5	4	3	2	1	0	
	9	0	1	1	1	0	0	0	0	3
	8	1	1	0	1	0	0	0	0	3
	7	0	0	0	0	0	1	1	1	3
	6	0	0	0	0	1	1	0	1	3
	5	1	1	1	1	0	0	0	1	5
	4	0	0	1	0	1	1	1	1	5
6		0	1	0	0	0	0	0	0	1
1		0	0	0	0	0	0	1	0	1
	3	1	0	0	0	0	0	1	1	3
	2	0	1	0	0	0	0	1	1	3
	1	0	0	1	0	0	0	1	1	3
	0	0	0	0	1	0	0	1	1	3

The device calculates the ECC by taking the EVEN parity of ECC check codes of all logic data bits. For example, if the 64 bit data is 0x45, the binary equivalent is 01000101. From Table 6, the required check codes are 00001101 (bit [6]), 01000011 (bit [2]) and 00010011 (bit [0]). Bitwise XOR of the check codes (even parity) results in an ECC value of 01011101.

If the device is configured to 32-bit mode, it reads 32 bits of data, pads bits [63:32] with zeros to create a 64-bit bus, and adds 8-bits of ECC data. When the device is configured to 16-bit mode, it reads 16 bits of data, pads bits [63:16] with zeros to create a 64-bit bus, and adds 8-bits of ECC data.The device calculates the ECC based on the 64-bit data and then compares that against the received ECC. The result of this comparison (bitwise XOR between received ECC and calculated ECC) is called the syndrome.

If the syndrome is 00000000, both the received data and ECC are correct.

- If the syndrome is any other value, the device assumes that there is an error either in the received data or the received ECC.
- If the syndrome contains a single 1, there is a single bit error in the ECC byte. For example, if the received data is 0x45, the calculated ECC is 01011101. If the received ECC is 01010101, the resulting syndrome is 00001000. Table 6 shows that this syndrome corresponds to check bit 3. Since there is no error in the data itself (just in the ECC byte), data correction is not necessary.
- If the result syndrome contains 2 or more 1’s, this indicates that there is a double-bit error.

Copyright © 2015 Marvell

5.1.2.2 PCI Express Data Integrity

The device supports the different PCI Express layers data protection mechanisms as defined in the PCI Express specification.

When the MAC layer detects a receiver error (for example, 8b/10b error), it drops the symbol and reports an error. When the link layer detects an error (for example, LCRC or sequence ID error), it drops the packet and reports an error. It also responds with acknowledge.

MAC and link layer errors are considered correctable errors; the initiator should re-transmit the packets. Still, If it is a repeated error, the link eventually enters a recovery state.

Until this point, when the transaction layer detects a poisoned TLP (EP bit set), it reports an error. The transaction is forwarded to the target interface with an erroneous data indication, except for the case of write to the device's internal registers, in which data is discarded.

Note The device does not support ECRC.

On the Tx side, the device always drives correct LCRC (and, in case of LCRC error in the receive side, retransmits).

5.1.3 Errors Forwarding

Although each interface includes the required logic to detect and report data errors, this is sometimes inadequate due to the latency of interrupt routines. For example, poisoned TLP is detected upon a PCI Express write to SDRAM. If the core processor reads this data from memory (without knowing that this is erroneous data), it can cause undesired results.

To guarantee that this scenario does not occur, the device supports an error forwarding mechanism.

When a transaction received on one interface is detected as erroneous:

- An error interrupt is asserted, and transaction information is registered.
- The transaction is forwarded to the target interface with an erroneous data indication.

The device supports an error-forwarding mechanism. When a transaction received on a unit or interface is detected as erroneous[1] (for example, poisoned TLP is detected upon a PCI Express write to SDRAM):

- An error interrupt is asserted, and transaction information is registered.
- The transaction is forwarded to the target interface with an erroneous data indication.

The device internal DMAs (XOR DMA, Ethernet SDMA, SATA DMA, SDIO DMA, USB DMA, TDM DMA) never generate erroneous data indication. However, they are sensitive to erroneous data indication received from other units:

- Ethernet SDMA, if it detects an erroneous data indication on data buffer read, transmit this buffer with bad CRC.
- Any of the DMAs stop if they encounter an erroneous data indication during descriptor read.

Note A write is discarded (register is not updated) when the device detects an error during a write to one of the chip internal registers.

1. For interfaces and units that support error forwarding

Copyright © 2015 Marvell

- <PerrProp> = 1 and <ECC> = 1: Forward error to DRAM—meaning, perform the RMW as usual, but flip ECC bits before the write, resulting in a non-correctable error in DRAM.
- <PerrProp> = 1 and <ECC> = 0: Data is discarded (write with all byte enables in-active).

5.1.3.1 Device Bus

The device does not support parity on the local device bus. In case of a write access to the device bus with bad parity indication, the device controller discards the data.

5.1.3.2 Core Processor

If a CPU read from the SDRAM results in an uncorrectable error, the error is propagated to the Marvell® Core Processor pipeline (transaction abort), and the Marvell® Core Processors enters an exception.

Similarly, if the <IO Err Prop Enable> field in the IO Error Control Register (Table 163 p. 466) is set to 1, upon a CPU read from the I/O device that results in erroneous read data, the error is propagated to the CPU core pipeline (transaction abort), and the core processor enters an exception.

5.1.3.3 PCI Express

If the <RxDPPropEn> field in the PCI Express Mbus Adapter Control Register (Table 445 p. 626) is set to 1, when receive poisoned TLPs (TLP with EP bit set), the PCI Express port forward the transaction to the target interface with erroneous data indication.

If the <TxDPPropEn> field in the PCI Express Mbus Adapter Control Register (Table 445 p. 625) is set to 1, the PCI Express port will force poisoned TLP (set EP bit) in case of:

- A PCI Express read from DRAM that results in none correctable ECC error
- Any case of a CPU or DMA write to PCI Express with erroneous data indication

5.1.4 Internal Data Path Parity Support

Each of the device internal buffers implemented as an SRAM array is parity protected. Each entry of SRAM is coupled with a parity bit (bit per buffer entry). Even parity is generated with every write to the SRAM and checked with every read from SRAM. If a parity error is detected, an interrupt is asserted (if not masked). For a list of exceptions, see Section 7.2, Interrupt Controller, on page 74.



Note The internal data path error cannot be related to a specific transaction. This means that it is very hard for the device to recover from this error. Typically, a data path error requires a reset.

For software development and debug purposes, parity generation can be set to ODD parity. This results in parity error detection (since checking is always for EVEN parity).

5.2 PCI Express Reference Clock Output

The device supports 2 pairs of PCI Express clock outputs. The direction of each clock pair is controlled through a dedicated reset strapping pin.

- Clock pair 0 is associated with PCIe port 0 that can be operated in Root Complex or Endpoint mode and with x1 or x4 lane width.
- Clock pair 1 is associated with PCIe ports 1, 2, and 3 and operates only in Root Complex mode and x1.

Each clock pair can be turned off dynamically for power saving. The device supports CLKREQn signalling as defined by the PCI Express specifications.

This functionality is available only when the port is configured as Root Complex.

For additional information about the PCIe physical layer configuration options see the device Hardware Specifications.

5.3 Big and Little Endian Support

The device supports both Big Endian and Little Endian byte ordering, as defined in the *ARM Architecture Reference Manual,* Second Edition. It also supports hardware features for performing data conversion on some of its interfaces.

5.3.1 CPU Core Byte Ordering

The Exception Endian mode is set by <EE> bit in register C1, within the CPU-CP15 registers. The value of the <EE> bit is reflected by the <CPU Big Endian Status> field in the CPU Control and Status Register (N=0–1) (Table 217 p. 494).

The initial value of the <EE> bit for CPU0 is set in the device reset strap.

The initial value of <EE> bit for CPU1 is set by the <Endian Init> field in the CPU Configuration Register (N=0–1) (Table 216 p. 494). The initial value of the <Endian Init> field is defined as Little Endian mode (0).

The CPU performs the proper data swapping, according to this setting. For full details, see the ARM *Cortex™-A9 MPCore® Technical Reference Manual*.



Note Regardless of the endianness mode, the device internal registers and DMAs descriptors always operate in Little Endian mode. When working in Big Endian mode, use a software macro to perform byte swapping when an internal register or a descriptor is written, or read.

5.3.2 PCI Express Space

The PCI Express specification defines that a TLP data payload be presented as Big Endian, and the packet header (address, command) be presented as Little Endian. The PCI Express interface meets these requirements. Depending on the chip endianness configuration, the PCI Express interface either performs or does not perform byte swapping.

The least significant byte is always the one to be transmitted first. For example, if a CPU is configured to Big Endian mode and there is a CPU write of 4 bytes to address 0x0 (which appears on the CPU bus as 64'hAABBCCDD.XXXXXXXX), the device drives 0xAA as the first byte on the PCI Express link.



Note The PCI Express interface does not support different byte swapping on a per master and slave operation, nor on a per address-window basis.

5.3.3 DMA Data Swapping

The device DMAs (XOR DMA, Buffer Manager DMA, USB DMA, TDM DMA, SATAHC DMA, and encryption DMA) support the required mechanisms for proper data transfer both in Big Endian and Little Endian environments.

Document Classification: Proprietary Information

5.3.3.1 XOR DMA Data Swapping

By default, the XOR DMA does not need to perform any data byte swapping when transferring data buffers over the Mbus.

However, the XOR DMA supports byte swapping on a 64-bit Qword basis upon a read from source buffers or a write to a destination buffer via the <DrdResSwp> field and the <DwrReqSwp> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371). This can be useful for endianness conversion.

The XOR DMA descriptor is fetched from memory as either a 32-byte or 64-byte burst, depending on the XOR mode of operation. The descriptor is loaded into the XOR DMA register file, which maintains the Little Endian convention. Thus, the descriptor is expected to be prepared in memory accordingly.

If the <DesSwp> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371) is set to 1, the DMA performs a byte swap on a 64-bit Qword basis when fetching (and closing) descriptors.

5.3.3.2 Cryptographic Engine Data Swapping

The device Cryptographic Engine and Security Accelerator (CESA) supports the following byte swap mechanisms:

- Byte swap and word swap (within an 8-byte Qword) upon read/write access to the CESA integrated SRAM. Set the Target Attributes field in the CPU address decoding registers to achieve the required data swapping (see Section 4.2, Cortex™-A9 Processor Address Decoding, on page 45).
- Byte swap (within an 8-byte Qword) upon a TDMA read/write access to memory. Clear the <BS> field in the Control Register (p=0–1) (Table 2426 p. 2268) to enable byte swapping.



Note

The data swapping logic cannot distinguish between descriptor and packet data. It is a static setting.

5.4 Spread Spectrum Clock Generator (SSCG)

This device incorporates a Spread Spectrum Clock Generator (SSCG) to reduce system EMI emission. The SSCG is applied to the CPU/L2 cache clock and the DDR clock.

The Spread Spectrum is characterized by two parameters:

- Modulation frequency
- Deviation (band)

The standard value for these parameters are a modulation frequency of 30 kHz.

5.4.1 Configuring the SSCG

Use the SSCG Frequency Configuration Register (Table 1912 p. 1411) to configure the Spread Spectrum Clock (SSC). The value of the SSCG Low Boundary should not be changed.



Note

Configure the SSCG before enabling it.

5.4.2 Enabling the SSCG

Set the reset strapping as shown in the Pin Sample Configuration section of the device Hardware Specifications.

To enable the SSCG, configure the SSCG SSC Mode Register (Table 1913 p. 1411) as follows:

1. De-assert the <SSC Reset> field and the <External PI Reset> field.
2. Wait at least 1 µs.
3. Set <PI Loop Control> to 0x1 (Enable).
4. Wait at least 1 µs.
5. Set <SSC Clock Enable> to 0x1 (Enable).

5.5 Integrated Thermal Sensor

The device includes an integrated thermal sensor for junction temperature monitoring.

The thermal sensor is an Analog-to-Digital Converter (A2D) that also performs automatic calibration and time average functions.

The thermal sensor may be polled at periodic time intervals by the host CPU to monitor the device junction temperature and take measures if it has stepped out of the target thermal envelope (the description of these measures is beyond the scope of this document).

For details on thermal management, refer to Section 28.6, Thermal Management.

Copyright © 2015 Marvell

6 Boot Flow

This section describes the device boot flow.

6.1 Features

The device BootROM:

- Executes according to the Sample at Reset and embedded eFuse configuration
- Performs basic initialization of the device
- Loads and executes code from one of the following boot devices:
  - SPI NOR Flash
  - NOR Flash
  - SPI NAND Flash
  - NAND Flash
  - SATA interface
  - PCI Express interface
  - SDIO interface
  - UART interface (using the Xmodem protocol)

Copyright © 2015 Marvell

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU
System on Chip

Functional Specifications – Unrestricted

Part 2: CPU Subsystem

- Section 7, CPU Subsystem
- Section 8, Timers, Counters, and Watchdog

Doc. No. MV-S109094-U0, Rev. A
November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell

7 CPU Subsystem

The device contains a CPU processing subsystem composed of:

- 2 ARMv7-A Cortex™-A9 32-bit processors in 88F6821, 88F6W21, 88F6820 and 88F6828
  or
  1 ARMv7-A Cortex™-A9 32-bit processor in 88F6810 and in 88F6811
- A Snoop Control Unit (SCU) for Symmetric Multiprocessing (SMP) interoperability
- A Layer 2 (L2) shared cache with a capacity of up to 1 MB (see Figure 9).

For full details and specifications of the Cortex-A9 CPU core, and the accompanying L2 cache, refer to the following ARM® documents (for additional details about these documents, see Related Documents on page 20).[1]

- *Cortex™-A9 Core Technical Reference Manual,* Revision r4p1
- *Cortex™-A9 MPCore® Technical Reference Manual,* Revision r4p1
- *Cortex™-A9 Floating-Point Unit Technical Reference Manual,* Revision r4p1
- *Cortex™-A9 NEON™ Media Processing Engine Technical Reference Manual,* Revision r4p1
- *Corelink™ Level 2 Cache Controller L2C-310 Technical Reference Manual,* Revision r3p3
- *ARM® Generic Interrupt Controller Architecture Specification* Architecture Version 1.0
- *Cortex™-A9 MBIST Controller Technical Reference Manual*, Revision r4p1
- *ARM® Architecture Reference Manual,* Second Edition

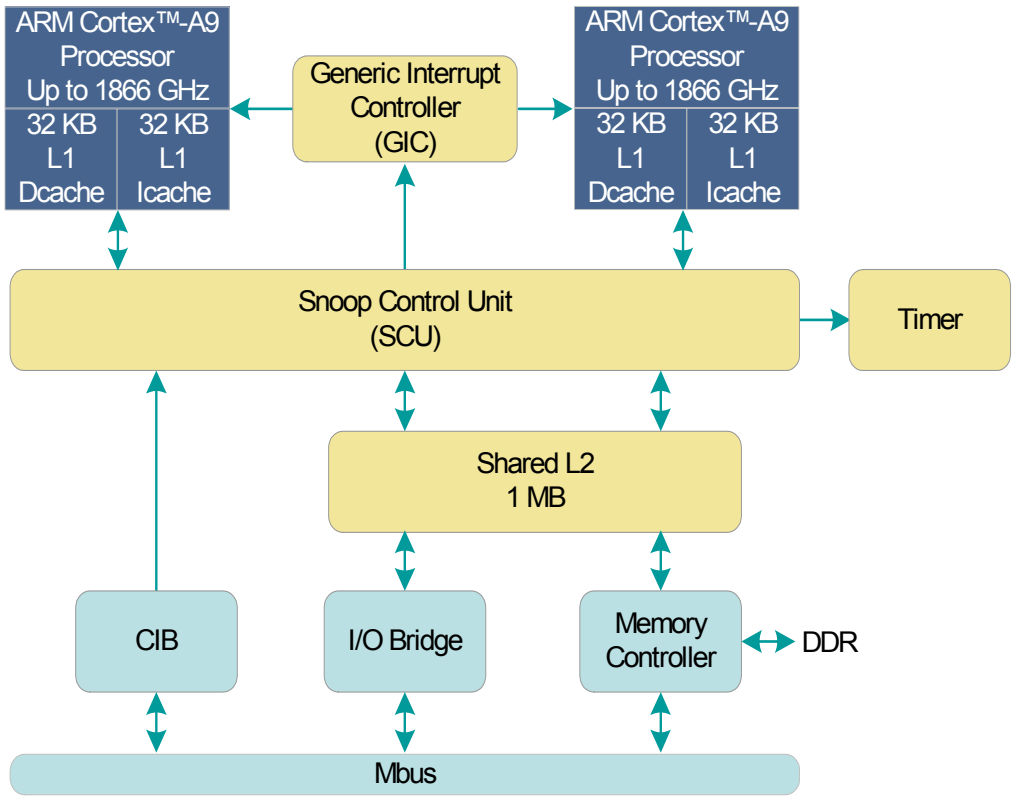
The CPU subsystem registers are located in Appendix A.2, CPU Subsystem Registers, on page 439.

Figure 9 presents the device CPU subsystem block diagram.

1. Some terms in this specification are registered trademarks or trademarks of ARM, including, but not exclusively, ARM, Corelink, Cortex, MPCore, and NEON.

Copyright © 2015 Marvell

Figure 9: Processor Subsystem Block Diagram



Note This section is written for the dual CPU device. Refer to Table 1, Device Differences, on page 34 to view the differences between the devices.

Copyright © 2015 Marvell

7.1 CPU Subsystem Features

7.1.1 Cortex™-A9 Feature Set

Each Cortex™-A9 processor includes the following features:

- Superscalar, variable length, out-of-order pipeline with dynamic branch prediction
- ARM architecture v7-A instruction set
- ARMv7-A Advanced SIMD and VFPv3 D-32 instruction sets extensions running over NEON Media Processing Engine (MPE)
- ARMv7-A Debug architecture
- Jazelle DBX (Direct Bytecode eXecution) support for Java execution
- Harvard level-1 (L1) cache:
  - Virtual Indexed Physically Tagged (VIPT) 4-way set-associative 32 KB Instruction cache
  - Physical Indexed Physically Tagged (PIPT) 4-way set-associative 32 KB Data cache
- Transaction Look-ahead Buffers (TLB) comprise of:
  - 64-entry fully associative Instruction micro TLB
  - 32-entry fully associative data micro TLB
  - 2-way-associative 4K entry unified TLB
- 2-level dynamic branch prediction with:
  - Global History Buffer (GHB) of 16K entries
  - Branch Target Address Cache (BTAC) of 4K
  - Return stack
- Data Preload Engine with an 8-entry FIFO
- Event Monitoring and Counting compliant to the Performance Monitor Unit (PMU) specifications
- Parity Error Detection for all RAMs in L1 Instruction cache, L1 Data cache, BTAC, CHG, and TLB

For further details about the Cortex-A9 processor functional specifications see the *Cortex-A9 Core Technical Reference Manual.*

7.1.2 Snoop Control Unit (SCU) Feature Set

The 2 processors interact with external memory, using a Snoop Control Unit (SCU) that contains the following features:

- Data cache coherency
- Asymmetric Multiprocessor Mode (AMP) in which data coherency is not maintained by hardware
- I/O Cache Coherency (IOCC) in which memory transactions initiated by non-processor elements do not violate L1 Data cache coherency
- Arbitration between Cortex-A9 processors and I/O coherent requesting L2 accesses
- Speculative Line-fills to the shared L2 cache

For further details of SCU functional specification, see the *Cortex-A9 MPCore Technical Reference Manual.*

Copyright © 2015 Marvell

7.1.3 Shared L2 Cache Feature Set

The 2 processors share a 1-MB, L2 cache that contains the following features:

- Physically Indexed Physically Tagged (PIPT)
- 16-way set-associative
- Line size of 32B
- Lockdown format C, for instruction and data
- Lockdown by Line
- Write policy: Write-Through (WT) and Write-Back (WB) modes as identified by the MMU outer-cache attribute
- Allocation policy: Read allocate (RA) and Write allocate (WA) as identified by the MMU outer-cache attribute
- Force write allocate option to always have cacheable writes allocated to L2 cache
- Parity Error Protection on Tag RAMs
- Critical Word First Line-fill support
- Pseudo-random and round-robin replacement policies
- Configurable mode for an exclusive L2 cache
- In-flight capacity:
  - Four 256-bit Line Fill Buffers (LFBs)
  - Two 256-bit Line Read Buffers (LRBs) for each slave port
  - Three 256-bit Eviction Buffers (EBs)
  - Three 256-bit Store Buffers (STBs)

For further details of the shared L2 functional specification see the *Corelink™ Level 2 Cache Controller L2C-310 Technical Reference Manual*.

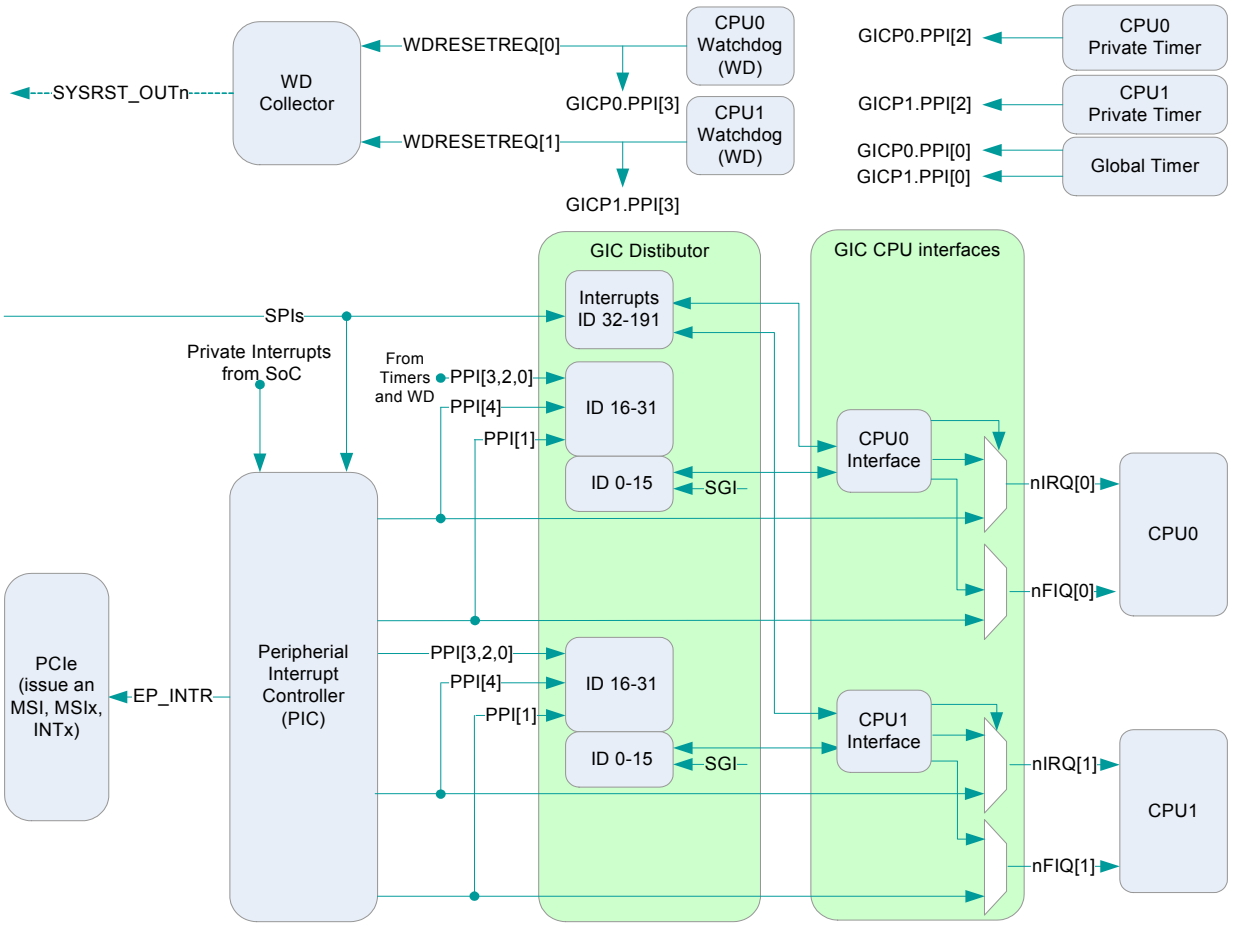
7.2 Interrupt Controller

The device contains a Multiprocessor interrupt controller for handling asynchronous interrupts in varying exception modes. The Interrupt Controller is compliant with the *ARM® Generic Interrupt Controller Architecture Specification Architecture version 1.0* (GIC v1.0). This section describes the main features of this interrupt controller. It does not reproduce information defined in the *ARM® Generic Interrupt Controller Architecture Specification Architecture version 1.0* or in the *Cortex™-A9 MPCore® Technical Reference Manual.* This section specifies the IMPLEMENTATION_DEFINES options for those specifications.

Copyright © 2015 Marvell

The multiprocessor interrupt controller flow appears in Figure 10.

Figure 10: Multiprocessor Interrupt Controller Flow



7.2.1 Feature Set

The interrupt controller access is memory map based, as defined for the GIC in Table 3, Device Internal Registers Address Map, on page 756. Details of the registers and their functionality can be found in the specifications mentioned earlier.

The Multiprocessor interrupt controller integrates the following features:

- Configurable model for multiprocessor interrupts distribution and prioritization
- Centralized Distribution engine for all the processors
- Local Delivery Interface for each processor
- Programmable mapping per interrupt sources to processor or to group of processors
- Programmable interrupt prioritization levels with a preemption mechanism
- Programmable Task Priority per processor
- Supports interrupt selection by maximal priority
- Generation and broadcasting of up to 16 standard ARM software generated interrupts (SGI)

- Generation and broadcasting of up to additional 32 proprietary software generated interrupts between I/O masters and the processors
- Generation and support of up to 6 standard ARM private peripheral interrupts (PPIs)
- Generation and support of up to 32 SoC private interrupts
- Generation and support of up to 160 shared peripherals interrupts
- Supports ARM Security Extension, setting each interrupt as either Secured or Non-secured and managing CPU accesses accordingly
- Global SoC Error Cause with per-processor masking
- PCIe Root Complex MSI capturing and delivering to the processors
- PCIe Endpoint (EP) interrupt generation and delivering up the link through MSI or legacy INTx
- Supplies mapped memory registers interface, accessed by all the CPUs

7.2.2 Interrupt Sources IDs

The Device Interrupt Controller supports up to 192 main interrupt sources. Each interrupt source is identified by an interrupt ID.

The 192 interrupts are divided to the following 3 groups:

- SGI—16 software generated interrupt, located at ID0–ID15.
- PPI—16 private peripheral interrupts, located at ID16–ID31.

  NOTE: ID28 and ID31 are PPIs that reflect a summary of the SoC private peripherals interrupts.
- SPI—160 shared peripheral interrupts, located at ID32–ID191

The interrupt sources and their mapping are listed in Table 7, Distributer Interrupt Map, on page 78 .

7.2.3 SoC Private Peripheral Interrupts

ID28 and ID31 are PPIs that reflect a summary of the SoC private peripheral interrupts.

Once an IRQ is asserted, and ID28 is identified as the source ID, an interrupt controller of the interrupted CPU can identify the specific SoC private interrupts that are pending service by reading the SoC PPI Cause Reg Register (N=0–1) (Table 229 p. 499).This register is banked per processor and as such the vector read is associated to the SoC PPIs of the interrupted CPU.

- A specific SoC Peripheral Interrupt can be masked by writing its ID to <SoC PPI ISM>
- A specific SoC Peripheral Interrupt can be unmasked by writing its ID to <SoC PPI ICM>

The SoC PPIs and their mapping are listed in Table 8, Proprietary Interrupt Controller Interrupts, on page 86.

7.2.4 SoC Error Reporting

The device Interrupt Controller implements a centralized cause registers that latches all error events. The latched SoC errors are observed through the SoC Main Interrupt Error 0 Cause Register (Table 207 p. 489) and SoC Main Interrupt Error 1 Cause Register (Table 208 p. 489).

These events are summed into a single interrupt that is banked for CPU0, CPU1, and, in case the device is configured as a PCIe endpoint device, for the PCIe Host device.

Copyright © 2015 Marvell

For each of these processing elements (CPU0, CPU1, and PCIe RC), there is a set of masks that enables a static configuration that defines for each error event, the processing element that will be interrupted as a result of its occurrence. For the CPUs, SoC Main Interrupt Error 0 Mask Register (N=0–1) (Table 232 p. 500) and SoC Main Interrupt Error 1 Mask Register (N=0–1) (Table 233 p. 501) define these setting. For the PCIe Host, Endpoint SoC Main Interrupt Error 0 Mask Register (Table 204 p. 487) and Endpoint SoC Main Interrupt Error 1 Mask Register (Table 205 p. 487) define these settings.

For each of these processing elements (CPU0, CPU1, and PCIe RC), ID4 of the SoC PPI is asserted when a non-masked event that occurs. For the CPU's this interrupt will be reflected as interrupt ID31 (PPI in the GIC). For an endpoint device, such an event will trigger an interrupt towards the RC.

The device errors' sources and mapping are detailed in Table 9, Proprietary Interrupt Controller Errors, on page 91.

7.2.5 PCIe Root Complex MSI Capturing

The Device Interrupt Controller and PCIe unit provide support for PCIe Root Complex inbound Message Signaled Interrupt (MSI).

Using the proprietary Software Triggered Interrupt Register mechanism, the device can capture up to 128 different interrupts in the Inbound Doorbell registers. To activate this option the peer PCIe Device (EP) must set its MSI address to a value that points to the Software Triggered Interrupt Register (Table 206 p. 488). The MSI vector indicates the target CPU and the interrupt ID. There are 2 types of delivery schemes to MSI.

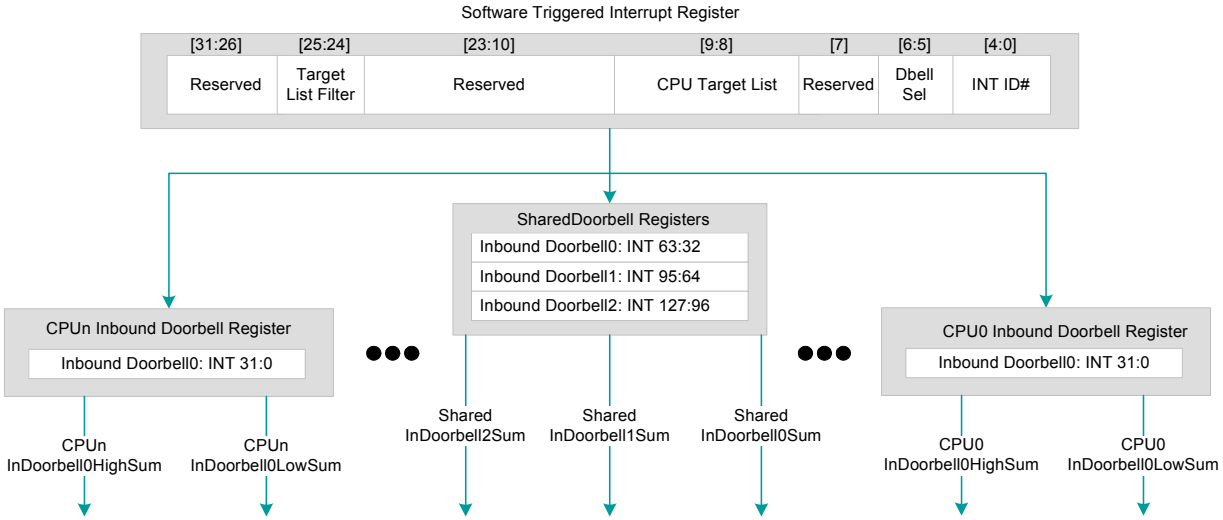
Private Interrupt: MSI vector[7:0] indicates on one of the interrupts 0–31. The interrupt will be send to all CPUs in the target list and all CPUs will serve the interrupt. The message asserts interrupt source ID28 or ID31. Since these are PPIs, they will be processed by each CPU independently. The interrupt handler can identify the message of the interrupt by examining the SoC Peripheral Private Interrupts and following that by identifying the set bit in the Inbound Doorbell Register (N=0–1) (Table 227 p. 499).

Shared Interrupts: MSI vector[7:0] indicates one of the interrupts 32–127. The interrupt will be send to all CPUs in the target list and only one CPU will serve the interrupt. The message asserts one of interrupt sources ID99–ID101. Since these interrupts are SPIs and as such will be handled by one of the CPUs, the remaining CPU will not be acknowledged (that is, it receives a spurious ID when reading from the GIC the ACKINTID in the ICCIAR). Following that, the interrupt handler can examine the corresponding MP Shared Inbound Doorbell registers (see Table 185–Table 187).

Copyright © 2015 Marvell

Figure 11 illustrates the entire Software Triggered Interrupt delivery scheme for serving inbound MSI/MSI-X messages.

Figure 11: MSI/MSI-X Capturing Scheme Using the Software Trigger Interrupt



7.2.6 PCIe Endpoint Interrupt Delivery

The device contains a separate mechanism for controlling and managing interrupt detection, and their propogation from the Device (as an EP) to its root complex.

For each interrupt, an EP interrupt mask field defines whether or not the assertion of the interrupt should be send to the root complex. This EP mask bit is located at field EP_Mask at the Endpoint Source i Control Register (i=29–115) (Table 212 p. 491).

7.2.7 Interrupts Mapping Tables

This section provides the following 3 mappings:

- Mapping of Interrupt source ID
- Mapping of SoC Peripheral Interrupts IDs
- Mapping of the SoC Errors

Table 7, Table 8, and Table 9 list the interrupt/error sources and their associated ID mapping. If the device does not contain a specific interface, the Interrupt ID is Reserved in that device.

Table 7: Distributer Interrupt Map

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
0	MP Internal	SGI[0]	Software generated interrupt
1	MP Internal	SGI[1]	Software generated interrupt
2	MP Internal	SGI[2]	Software generated interrupt
3	MP Internal	SGI[3]	Software generated interrupt

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
4	MP Internal	SGI[4]	Software generated interrupt
5	MP Internal	SGI[5]	Software generated interrupt
6	MP Internal	SGI[6]	Software generated interrupt
7	MP Internal	SGI[7]	Software generated interrupt
8	MP Internal	SGI[8]	Software generated interrupt
9	MP Internal	SGI[9]	Software generated interrupt
10	MP Internal	SGI[10]	Software generated interrupt
11	MP Internal	SGI[11]	Software generated interrupt
12	MP Internal	SGI[12]	Software generated interrupt
13	MP Internal	SGI[13]	Software generated interrupt
14	MP Internal	SGI[14]	Software generated interrupt
15	MP Internal	SGI[15]	Software generated interrupt
16	MP Internal	Reserved	
17	MP Internal	Reserved	
18	MP Internal	Reserved	
19	MP Internal	Reserved	
20	MP Internal	Reserved	
21	MP Internal	Reserved	
22	MP Internal	Reserved	
23	MP Internal	Reserved	
24	MP Internal	Reserved	
25	MP Internal	Reserved	
26	MP Internal	Reserved	
27	MP Internal	Global Timer	
28	MP Internal	PIC FIQ Summary	GIC recognizes the FIQ summary interrupt only when the FIQ is enabled in the GIC CPU interface (bit 3 in the CPU interface control register). When the FIQ enabled, even if the PIC interrupt is set to IRQ, GIC gets the FIQ summary interrupt.
29	MP Internal	Private Timer	
30	MP Internal	Watchdog Timer	

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
31	MP Internal	PIC IRQ Summary	
Shared Interrupts Sources			
32	0	Reserved	
33	1	SPI0 Interrupt	
34	2	I2C0 Interrupt	
35	3	I2C1 Interrupt	
36	4	Reserved	
37	5	Reserved	
38	6	Reserved	
39	7	Reserved	
40	8	Global Timer 0 interrupt	
41	9	Global Timer 1 interrupt	
42	10	Global Timer 2 interrupt	
43	11	Global Timer 3 interrupt	
44	12	UART0 interrupt	
45	13	UART1 interrupt	
46	14	Reserved	
47	15	Reserved	
48	16	USB0 interrupt	USB3 port 0 (Host) interrupt
49	17	USB1 interrupt	USB3 Port 1 (Host) interrupt
50	18	USB2 interrupt	USB2 (Host/Device) interrupt
51	19	CESA0 interrupt	
52	20	CESA1 interrupt	
53	21	RTC interrupt	
54	22	XOR channel0 interrupt	
55	23	XOR channel1 interrupt	
56	24	BM interrupt	
57	25	SDIO interrupt	
58	26	SATA0 interrupt	

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
59	27	TDM interrupt	
60	28	SATA1 interrupt	
61	29	PCIe0 interrupt	PCI Express Port0 INTA/B/C/D assert message interrupt.
62	30	Reserved	
63	31	Reserved	
64	32	Reserved	
65	33	PCIe1 interrupt	PCI Express Port1 INTA/B/C/D assert message interrupt.
66	34	Reserved	
67	35	Reserved	
68	36	Reserved	
69	37	GbE0 interrupt summary	
70	38	GbE0 interrupt Rx	
71	39	GbE0 interrupt Tx	
72	40	GbE0 interrupt Misc	
73	41	GbE1 interrupt summary	
74	42	GbE1 interrupt Rx	
75	43	GbE1 interrupt Tx	
76	44	GbE1 interrupt Misc	
77	45	GbE2 interrupt summary	
78	46	GbE2 interrupt Rx	
79	47	GbE2 interrupt Tx	
80	48	GbE2 interrupt Misc	
81	49	Reserved	
82	50	Reserved	
83	51	Reserved	
84	52	Reserved	
85	53	GPIO[7:0] interrupt	
86	54	GPIO[15:8] interrupt	
87	55	GPIO[23:16] interrupt	

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
88	56	GPIO[31:24] interrupt	
89	57	Reserved	
90	58	GPIO[39:32] interrupt	
91	59	GPIO[47:40] interrupt	
92	60	GPIO[55:48] interrupt	
93	61	GPIO[59:56] interrupt	
94	62	Reserved	
95	63	SPI1 Interrupt	
96	64	Watchdog	
97	65	XOR channel 2 interrupt	
98	66	XOR channel 3 interrupt	
99	67	Shared Inbound Doorbell0	MSIx shared interrupt 0
100	68	Shared Inbound Doorbell1	MSIx shared interrupt 1
101	69	Shared Inbound Doorbell2	MSIx shared interrupt 2
102	70	PCIe 2 interrupt	PCI Express Port2 INTA/B/C/D assert message interrupt.
103	71	PCIe 3 interrupt	PCI Express Port3 INTA/B/C/D assert message interrupt.
104	72	Reserved	
105	73	PMU interrupt	CPU0 PM interrupt
106	74	PMU interrupt	CPU1 PM interrupt
107	75	Audio interrupt	
108	76	USB3 interrupt	USB3 (Device) interrupt
109	77	Reserved	
110	78	DFX Server Interrupt <i>	
111	79	DRAM controller interrupt	For automatic power modes entry/exit indications
112	80	GbE0 Wake on LAN	
113	81	GbE1 Wake on LAN	
114	82	GbE2 Wake on LAN	

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
115	83	Reserved	
116	84	NAND Flash Controller	
117	85	Misc interrupts on the SoC level	
118	86	Reserved	
119	87	Reserved	
120	88	Reserved	
121	89	Reserved	
122	90	Reserved	
123	91	Reserved	
124	92	Reserved	
125	93	Reserved	
126	94	Reserved	
127	95	Reserved	
128	96	Reserved	
129	97	Reserved	
130	98	Reserved	
131	99	Reserved	
132	100	Reserved	
133	101	Reserved	
134	102	Reserved	
135	103	Reserved	
136	104	Reserved	
137	105	Reserved	
138	106	Reserved	
139	107	Reserved	
140	108	Reserved	
141	109	Reserved	
142	110	Reserved	

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
143	111	Reserved	
144	112	Reserved	
145	113	Reserved	
146	114	Reserved	
147	115	Reserved	
148	116	Reserved	
149	117	Reserved	
150	118	Reserved	
151	119	Reserved	
152	120	Reserved	
153	121	Reserved	
154	122	Reserved	
155	123	Reserved	
156	124	Reserved	
157	125	Reserved	
158	126	Reserved	
159	127	Reserved	
160	128	MSI Frame 0 Int [0]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
161	129	MSI Frame 0 Int [1]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
162	130	MSI Frame 0 Int [2]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
163	131	MSI Frame 0 Int [3]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
164	132	MSI Frame 0 Int [4]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
165	133	MSI Frame 0 Int [5]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
166	134	MSI Frame 0 Int [6]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
167	135	MSI Frame 0 Int [7]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
168	136	MSI Frame 0 Int [8]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
169	137	MSI Frame 0 Int [9]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
170	138	MSI Frame 0 Int [10]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
171	139	MSI Frame 0 Int [11]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
172	140	MSI Frame 0 Int [12]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
173	141	MSI Frame 0 Int [13]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
174	142	MSI Frame 0 Int [14]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
175	143	MSI Frame 0 Int [15]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
176	144	MSI Frame 1 Int [0]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
177	145	MSI Frame 1 Int [1]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
178	146	MSI Frame 1 Int [2]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
179	147	MSI Frame 1 Int [3]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
180	148	MSI Frame 1 Int [4]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
181	149	MSI Frame 1 Int [5]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
182	150	MSI Frame 1 Int [6]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
183	151	MSI Frame 1 Int [7]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
184	152	MSI Frame 1 Int [8]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
185	153	MSI Frame 1 Int [9]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0

Copyright © 2015 Marvell

Table 7: Distributer Interrupt Map (Continued)

Interrupt ID	SoC-to-MPIC Index	Mapping	Description
186	154	MSI Frame 1 Int [10]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
187	155	MSI Frame 1 Int [11]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
188	156	MSI Frame 1 Int [12]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
189	157	MSI Frame 1 Int [13]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
190	158	MSI Frame 1 Int [14]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0
191	159	MSI Frame 1 Int [15]	SoC Generic Interrupt Controller MSI Interrupt associated with frame group 0

Table 8: Proprietary Interrupt Controller Interrupts

Interrupt ID	SoC-to-MPIC index	Mapping	Description
0	MP Internal	Inbound Doorbell low summary	
1	MP Internal	Inbound Doorbell high summary	
2	MP Internal	Outbound Doorbell summary	
3	MP Internal	MP local sum	
4	MP Internal	SoC Error Summary	
5	MP Internal	Local Timer0	
6	MP Internal	Local Timer1	
7	MP Internal	Local Watchdog	
8	MP Internal	Port0 Th RXTX int	
9	MP Internal	Port0 RXTX int	
10	MP Internal	Port1 Th RXTX int	
11	MP Internal	Port1 RXTX int	
12	MP Internal	Port2 Th RXTX int	
13	MP Internal	Port2 RXTX int	
14	MP Internal	Reserved	

Copyright © 2015 Marvell

Table 8: Proprietary Interrupt Controller Interrupts (Continued)

Interrupt ID	SoC-to-MPIC index	Mapping	Description
15	MP Internal	Reserved	
16	MP Internal	CPU_GPIO_0_7_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[7:0]
17	MP Internal	CPU_GPIO_15_8_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[15:8]
18	MP Internal	CPU_GPIO_23_16_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[23:16]
19	MP Internal	CPU_GPIO_31_24_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[31:24]
20	MP Internal	CPU_GPIO_39_32_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[39:32]
21	MP Internal	CPU_GPIO_47_40_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[47:40]
22	MP Internal	CPU_GPIO_55_48_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[55:48]
23	MP Internal	CPU_GPIO_63_56_Int	Each CPU has a dedicated set of mask. level and interrupt cause registers that sum GPIO[59:56]
24	MP Internal	Reserved	
25	MP Internal	Reserved	
26	MP Internal	CNTHP_Timer_int	
27	MP Internal	CNTPNS_Timer_int	
28	MP Internal	CNTPS_Timer_int	
29	MP Internal	CNTV_Timer_int	
30	MP Internal	Reserved	
31	MP Internal	Reserved	
Shared Interrupts Sources			
32	0	Reserved	
33	1	SPI0 Interrupt	
34	2	I2C0 Interrupt	
35	3	I2C1 Interrupt	
36	4	Reserved	
37	5	Reserved	

Table 8: Proprietary Interrupt Controller Interrupts (Continued)

Interrupt ID	SoC-to-MPIC index	Mapping	Description
38	6	Reserved	
39	7	Reserved	
40	8	Global Timer 0 interrupt	
41	9	Global Timer 1 interrupt	
42	10	Global Timer 2 interrupt	
43	11	Global Timer 3 interrupt	
44	12	UART0 interrupt	
45	13	UART1 interrupt	
46	14	Reserved	
47	15	Reserved	
48	16	USB0 interrupt	USB3 0 (Host) interrupt
49	17	Reserved	
50	18	Reserved	
51	19	Reserved	
52	20	Reserved	
53	21	Reserved	
54	22	XOR channel0 interrupt	
55	23	XOR channel1 interrupt	
56	24	Reserved	
57	25	SDIO interrupt	
58	26	Reserved	
59	27	Reserved	
60	28	Reserved	
61	29	PCIe0 interrupt	PCI Express Port0 INTA/B/C/D assert message interrupt.
62	30	Reserved	
63	31	Reserved	
64	32	Reserved	
65	33	PCIe1 interrupt	PCI Express Port1 INTA/B/C/D assert message interrupt
66	34	Reserved	

Copyright © 2015 Marvell

Table 8: Proprietary Interrupt Controller Interrupts (Continued)

Interrupt ID	SoC-to-MPIC index	Mapping	Description
67	35	Reserved	
68	36	Reserved	
69	37	GbE0 interrupt summary	
70	38	GbE0 interrupt Rx	
71	39	GbE0 interrupt Tx	
72	40	GbE0 interrupt Misc	
73	41	GbE1 interrupt summary	
74	42	GbE1 interrupt Rx	
75	43	GbE1 interrupt Tx	
76	44	GbE1 interrupt Misc	
77	45	GbE2 interrupt summary	
78	46	GbE2 interrupt Rx	
79	47	GbE2 interrupt Tx	
80	48	GbE2 interrupt Misc	
81	49	Reserved	Reserved
82	50	Reserved	
83	51	Reserved	
84	52	Reserved	
85	53	GPIO[7:0] interrupt	
86	54	GPIO[15:8] interrupt	
87	55	GPIO[23:16] interrupt	
88	56	GPIO[31:24] interrupt	
89	57	Reserved	
90	58	GPIO[39:32] interrupt	
91	59	GPIO[47:40] interrupt	
92	60	GPIO[55:48] interrupt	
93	61	GPIO[59:56] interrupt	
94	62	Reserved	
95	63	SPI1 Interrupt	

Copyright © 2015 Marvell

Table 8: Proprietary Interrupt Controller Interrupts (Continued)

Interrupt ID	SoC-to-MPIC index	Mapping	Description
96	64	Watchdog	
97	65	XOR channel 2 interrupt	
98	66	XOR channel 3 interrupt	
99	67	Shared Inbound Doorbell0	MSIx shared interrupt 0
100	68	Shared Inbound Doorbell1	MSIx shared interrupt 1
101	69	Shared Inbound Doorbell2	MSIx shared interrupt 2
102	70	PCIe 2 interrupt	
103	71	PCIe 3 interrupt	
104	72	Reserved	
105	73	PMU interrupt 0	
106	74	PMU interrupt 1	Thermal overheat and cooling interrupt
107	75	Audio interrupt	
108	76	USB 3 interrupt	
109	77	Reserved	
110	78	DFX Server Interrupt	
111	79	DRAM controller interrupt	For automatic power modes entry/exit indications
112	80	GbE0 Wake on LAN interrupt	
113	81	GbE1 Wake on LAN interrupt	
114	82	GbE2 Wake on LAN interrupt	
115	83	Reserved	
116	84	NAND Flash Controller	
117	85	Misc interrupts on the SoC level	
118	86	Reserved	

Copyright © 2015 Marvell

Table 9: Proprietary Interrupt Controller Errors

Error ID	SoC-to-MPIC Index	Mapping	Description
0	0	CryptoErr	Cryptographic engine error summary
1	1	DevErr	Device bus error (DEV_READYn timer)
2	2	Reserved	
3	3	XOR1Err	
4	4	PCIe0Err	PCIe port0 error (summary of PCI Express Cause register)
5	5	PCIe1Err	PCIe port1 error (summary of PCI Express Cause register)
6	6	GbEErr	Gigabit Ethernet error (address decoding and protection) Summary of Ethernet Unit Interrupt Cause (EUIC) registers of all GbE ports.
7	7	Reserved	
8	8	USBErr	USB error (address decoding and RAM parity) Summary of errors from all USB ports
9	9	DRAMErr	DRAM ECC error
10	10	XOR0Err	XOR engine error
11	11	Reserved	
12	12	BMUErr	Buffer Management Unit error summary
13	13	Reserved	
14	14	N/A	
15	15	PEX2Err	
16	16	PEX3Err	
17	17	Reserved	
18	18	Reserved	
19	19	Reserved	
20	20	ADErr	Audio Unit error summary
21	21	NAND Flash Ctrl Err	NFC error summary
22	22	Parity Err	Parity error interrupt (see Misc Error Interrupt Cause Register)
23	23	L2C	L2C Decode error received on master port from L3
24	24	L2C	L2C Event Counter Overflow or Event Counter Increment

Copyright © 2015 Marvell

Table 9: Proprietary Interrupt Controller Errors (Continued)

Error ID	SoC-to-MPIC Index	Mapping	Description
25	25	L2C	L2C Error on L2 data RAM read
26	26	L2C	L2C Error on L2 tag RAM read
27	27	L2C	L2C Error on L2 data RAM write
28	28	L2C	L2C Error on L2 tag RAM write
29	29	L2C	L2C Combined Interrupt Output
30	30	L2C	L2C Parity error on L2 data RAM read
31	31	L2C	L2C Parity error on L2 tag RAM read
32	32	L2C	L2C Slave error received on master port from L3
33	33	CPU0	Data RAM parity error CPU0
34	34	CPU0	Data tag RAM parity error CPU0
35	35	CPU0	Data outer RAM parity error CPU0
36	36	CPU0	Main TLB parity error CPU0
37	37	CPU0	Instruction data RAM parity error CPU0
38	38	CPU0	Instruction tag RAM parity error CPU0
39	39	CPU0	GHB parity error CPU0
40	40	CPU0	BTAC parity error CPU0
41	41	CPU1	Data data RAM parity error CPU1
42	42	CPU1	Data tag RAM parity error CPU1
43	43	CPU1	Data outer RAM parity error CPU1
44	44	CPU1	Main TLB parity error CPU1
45	45	CPU1	Instruction data RAM parity error CPU1
46	46	CPU1	Instruction tag RAM parity error CPU1
47	47	CPU1	GHB parity error CPU1
48	48	CPU1	BTAC parity error CPU1
49	49	Reserved	
50	50	Reserved	
51	51	Reserved	
52	52	Reserved	
53	53	Reserved	
54	54	Reserved	

Copyright © 2015 Marvell

Table 9: Proprietary Interrupt Controller Errors (Continued)

Error ID	SoC-to-MPIC Index	Mapping	Description
55	55	Reserved	
56	56	Reserved	
57	57	Reserved	
58	58	Reserved	
59	59	Reserved	
60	60	Reserved	
61	61	Reserved	
62	62	Reserved	
63	63	Reserved	

Copyright © 2015 Marvell

7.3 I/O Coherency

7.3.1 The Concept

The device enables programmable IO coherency for all its interfaces. Each of the device units supports programmable DRAM address windows, in which cache coherency is maintained.

Each of the device units' initiated transactions through the DRAM has two attribute bits to determine the required coherency process for this transaction—*Shared* and *Level-2 Allocate* attributes:

Read and write requests can be Shared or Non-shared. The sharing mode is initiated by the device units, using the windows attribute for address decoding or implementation a specific mechanism for higher resolution classification of transactions to a shared range. Non-shared requests are routed directly to the DRAM, while shared requests are routed to the Coherent I/O Bridge for snooping the CPUs' caches and Level-2 cache.

Write to a shared address can be configured to perform Level-2 Cache Deposit, initiated by the device units, using the windows attribute for address decoding or implementing a specific mechanism for higher resolution classification of transactions to a shared range. When L2 Deposit is required, a full cache-line I/O write is written directly to the L2 cache, regardless of whether it is hit or miss. A partial write is written to the L2 cache only on hit.

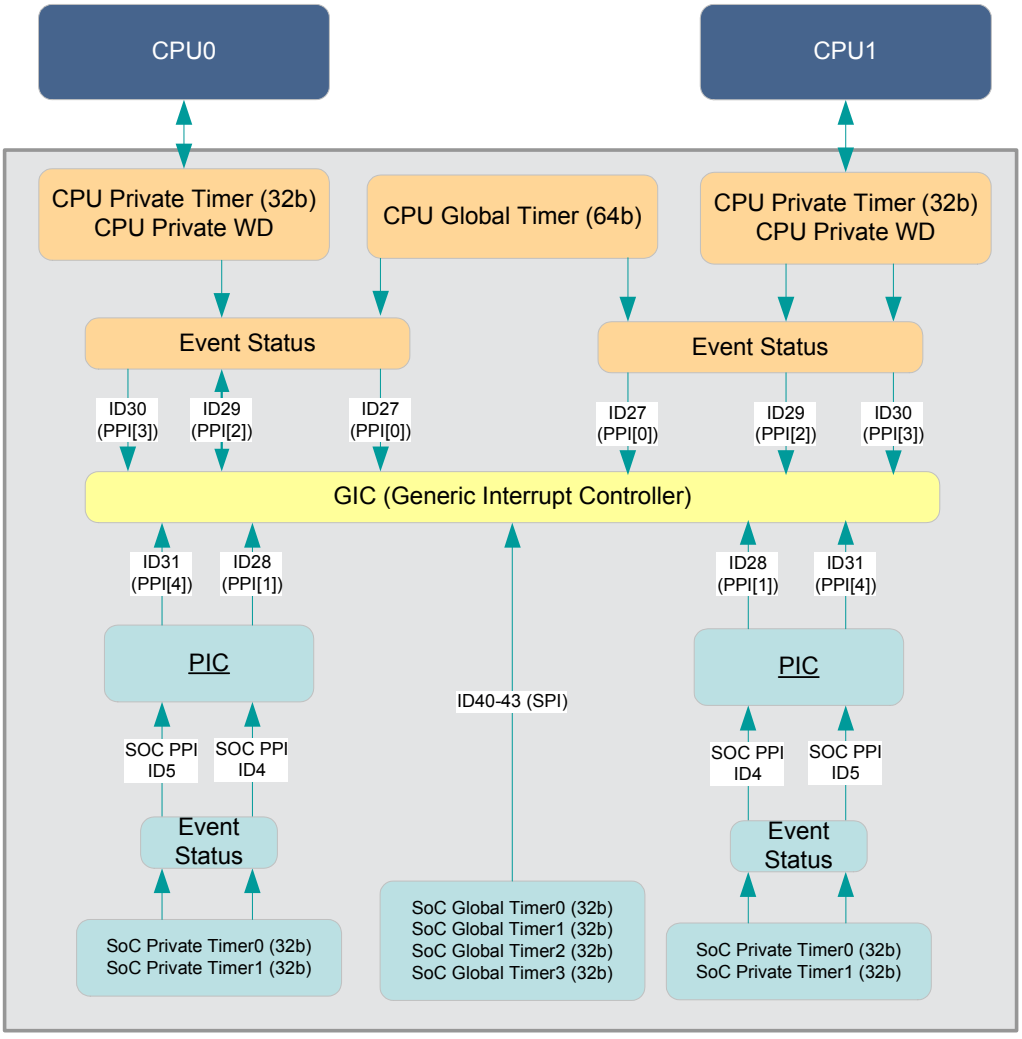
Copyright © 2015 Marvell

8 Timers, Counters, and Watchdog

This section describes the SoC global timers, private counters, and the configuration propagation options of Watchdog events. For information about the CPU private counters, CPU private watchdog and the global counter, refer to the *Cortex™-A9 Core Technical Reference Manual* (see Related Documents on page 20).

Figure 12 details the dual-CPU timers topology, including the private and global partitioning and events connectivity to the main interrupt controller.

Figure 12: Dual-CPU SoC Timers Topology



The registers related to the timers are located in Appendix A.2, CPU Subsystem Registers, on page 439.

Copyright © 2015 Marvell

8.1 Features

The device contains the following counters:

- One 64-bit CPU global timer
- One 32-bit CPU private timer per CPU (total of 2 private timers)
- One 32-bit CPU private watchdog per CPU (total of 2 watchdogs)
- Four 32-bit SoC global counters
- Two 32-bit SoC private counters (total of 4 SoC Private counters)

The single Global timer contains:

- 64-bit incrementing counter with auto-incrementing feature
- Memory mapped accesses
- Private compared value per CPU that asserts a private peripheral interrupt (PPI)

The SoC Global counters, SoC Private counters, and CPU Private timer integrate:

- A 32-bit counter that generates an interrupt when it reaches zero
- An 8-bit prescaler to enable better control of the period
- Configurable single-shot or auto-reload modes
- Configurable starting values for the counter

The CPU Watchdog timer contains:

- A 32-bit counter that generates a watchdog triggering event when it reaches zero
- Triggering event impact configurable to cause any of the following:
  - An interrupt to the CPU that is privately associated with the watchdog
  - An interrupt to the peer CPU
  - An interrupt to the PCIe host (when the device is configured as a PCIe Endpoint)
  - A system reset event
- An 8-bit prescaler to enable better control of the period
- Configurable single-shot or auto-reload modes
- Configurable starting values for the counter

8.2 Functional Description

This section describes the timer and counter modes of the SoC Global counters and the SoC Private counters.

8.2.1 Timer/Counter Modes

Each Timer/Counter delivers the following functionality.

Counter Mode — In the Counter mode, the counter counts down, stops when it reaches 0, and issues an interrupt

Setting the <CPUTimer*Auto> bit in each CPU Timer Control Register to 0 programs it to Counter mode

Timer Mode — In the Timer mode, the timer counts down, issues an interrupt when it reaches 0, reloads itself to the Reload register's value, and continues counting.

Setting the <CPUTimer*Auto> bit in each CPU Timer Control Register to 1 programs it to Timer mode

Each counter/timer has a Reload register and a Timer register. The Reload register represents the initial value to be loaded to the timer/counter. The Timer register represents the timer/counter itself. The software polls this register to determine the actual timer value.

Copyright © 2015 Marvell

Each CPU has the option to mask the interrupt asserted by a timer/counter expiration. This applies for both the SoC Global timers/counters and the Private CPU timers/counters.

For example, it is possible to assign SoC global timer 0 to CPU0 by masking the interrupt of this counter for CPU1. Similarly, it is possible to assign timer1 to CPU1 by masking its interrupt to CPU0.

8.2.2 SoC Global Timers/Counters

The 4 timers/counters:

- Are accessible and shared among all the CPUs.
- Each can be programmed to interrupt a specific CPU, or a group of CPUs, through the main interrupt controller with an appropriate priority. The main interrupt controller delivers the interrupt to the attached CPUs, with respect to the running priority.

The SoC global set is controlled through the following registers:

SoC Global Timers Control Register	Includes all the timers/counters enable and control functions. This includes a per counter enable, mode of operation, and an actual counter updating period relative to its source clock.
SoC Global Timers Events Status register Register	Summarizes the expiration status for all of the timers. This is an RW0C type register. A corresponding interrupt is asserted to the CPUs, if enabled in the main interrupt controller. The interrupt handler de-asserts the active SoC global timer interrupt line by clearing the corresponding timer expiration event in this register.
SoC Global Timer 0 Reload Register and **SoC Global Timer 0 Register**	The timer 0 register couple. The same set exists for the other SoC global timers and watchdog timer.

8.2.3 Private Timers/Counters

The private set includes 2 timers/counters and 1 watchdog timer per CPU.

- Each CPU timer set is placed in different register banks. It is virtually accessed with the same address from the CPU it is tied to.
- All the CPUs private timers sets are independent, with respect to each other, and expiration event interrupts only the CPU it belongs to through the main interrupt controller private interrupt lines. Timer expiration events are virtually mapped to same interrupt line, for example. CPU0 timer0 expiration asserts CPU0 private interrupt line ID and only its interrupt, while CPU1 timer1 expiration asserts same interrupt line ID but to CPU1 only.

Each CPU private set is controlled through the following registers:

SoC Private Timers Control Register (N=0–1)	Includes all the timers/counters enable and control functions. This includes a per counter enable, mode of operation, and an actual counter updating period relative to its source clock.
SoC Private Timers Events Status Register (N=0–1)	Summarizes each CPU timers expiration status. This is an RW0C type register. A corresponding interrupt is asserted to the specified CPU, if enabled in the main interrupt controller. The interrupt handler de-asserts the active SoC private timer interrupt line by clearing the corresponding timer expiration event in this register.
SoC Private Timer 0 Reload Register (N=0–1) and **SoC Global Timer 0 Reload Register**	The CPU Timer 0 register couple. The same set also exists for Timer 1 and the Watchdog timer.

8.2.4 Watchdog Reset

For each CPU, the device includes a private watchdog timer and a global watchdog timer that can be independently activated, reloaded, and expired. The device delivers flexibly programmable options for mapping the different watchdog timers expiration events to single system watchdog reset event.

The following registers control the generation of watchdog expiration reset.

The <CPU WD Group> field in the Watchdog (WD) RSTOUTn Mask Register (Table 195 p. 484) defines the CPUs that participate in the watchdog group. A watchdog reset is generated on the timers expiration of all the group members.

In addition to this mechanism, every CPU and the global watchdog has an independent mask bit. Once set, a reset is generated upon expiration, independent to the group value. This is controlled through programming of <CPU0 WD Mask> and <Global WD Mask> field in the Watchdog (WD) RSTOUTn Mask Register (Table 195 p. 484).

For example, when a watchdog reset is required on expiration of the global watchdog timer or both CPU0 and CPU1 private watchdog timers, enable the <Global WD Mask> and program the <CPU WD Group> to include both CPU0 and CPU1.

Invoking the Watchdog expiration reset is controlled by programming the <CPU0 WD Mask> and <Global WD Mask> field in the Watchdog (WD) RSTOUTn Mask Register (Table 195 p. 484).

8.2.5 Watchdog Interrupt

Watchdog timers expiration can be configured to generate an interrupt.

The <CPUn Local WD Interrupt Mask> field in the Watchdog (WD) Interrupt Mask Register (Table 196 p. 484) defines for each CPU which of the Private Watchdog timers expiration generates an interrupt to the corresponding CPU.

For example, when the expiration of the Private Watchdog of CPU0 requires an interrupt to CPU1 and the expiration of the Private Watchdog of CPU1 requires an interrupt to CPU0, set bit[1] in the <CPU0 Local WD Interrupt Mask> and set bit[0] in the <CPU1 Local WD Interrupt Mask>.

Copyright © 2015 Marvell



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU
System on Chip

Functional Specifications – Unrestricted

Part 3: External Memory Interfaces

- Section 9, DRAM Memory Controller
- Section 10, NAND Flash Controller (NFC)
- Section 11, Device Bus Controller

Doc. No. MV-S109094-U0, Rev. A
November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell

9 DRAM Memory Controller

This section describes the device DRAM memory controller. The controller is fully compliant with the JEDEC specification with respect to supported DRAM protocols and topologies.

Note The Marvell® proprietary DDR link training algorithm enabling higher DRAM clock speed is beyond the scope of this specification. For more information, contact a Marvell representative.

9.1 Feature List

The device DRAM memory controller supports the following features:

- DDR3/DDR3L and DDR4 Protocols:
  - The 88F6810, 88F6820, 88F6828 support DDR3, DDR3L and DDR4 protocols with configurable 32/16-bits data widths and optional ECC support.
  - The 88F6811, 88F6821, 88F6W21 support DDR3 and DDR3L protocols with 16-bits data widths and optional ECC support.
- SSTL 1.5V/1.35V I/O for DDR3 and 1.2V POD I/O for DDR4
- Up to 4 physical ranks (Chip Selects)
  - The 88F6810, 88F6820, 88F6828 support 2 Chip Selects.
  - The 88F6811, 88F6821, 88F6W21 support a single Chip Select.
- Up to 8Gb DDR3/DDR4 component densities respectively
- Bank interleaving between up to 16/32 open pages for DDR3/DDR4 protocols respectively
- 2T and 3T modes for heavy-load topologies
- Write and Read Leveling
- Multiple system to DRAM physical address mapping
- Extensive power saving options, such as Self Refresh, Pre-Charged Power Down and Active Power Down.
- Address mirroring
- Enhanced, configurable smart transaction scheduling for increased bus utilization
- Automatic calibration of I/Os impedance

The device does not support the following optional DDR3 features:

- Fixed BL4 and BC4 (only BL8 is supported)
- Additive latency
- TDQS

The device does not support the following optional DDR4 features:

- Fixed BL4 and BC4 (only BL8 is supported)
- Additive latency
- Write Data Bus Inversion (DBI)
- Write CRC
- C/A parity protection
- Control Gear-down mode
- Programmable preamble (2CK)

Copyright © 2015 Marvell

- 3DS
- Fine Granularity refresh
- CAL (Command Address Latency)
- 2tCK write preamble
- TDQS



Note

The ARMADA® 38x Family does not support DIMM devices.

Copyright © 2015 Marvell

9.2 DDR4 Features

The controller supports DDR4 devices.

9.2.1 MPR Read

To read the MPR register, perform these steps:



Note

Be aware of the DQ swapping done at the board level. This information can be derived from the outcome of the training sequence.

To enter the MPR Read mode:

1. Flush DLB.
2. Invalidate DLB valid lines.
3. Disable DLB.
4. Configure the DRAM to MPR mode using the DDR4_MRS3 OpCmd and configure MPR parameters, such as page and read format (see the DDR4 MR3 CSn Register (n=0–1) (Table 370 p. 713) and the SDRAM Operation Register (Table 290 p. 641)).
5. Access the MPR registers, using external access with following guidelines:
   a) Set the MPR location in the read address in the bank bits[1:0] location according to the address decoding tables.
   b) Set 3'b000 in the read address in the COL bits[2:0] location according to the address decoding table (JEDEC instructs A[2:0]=3'b000).
   c) Determine a read size that results in a single CAS command on DRAM (the size depends on the DRAM width).
6. In case of ECC Enable, ignore the ECC error indication since the ECC module will return its own MPR data and not the ECC calculation.
7. To read MPR of the ECC module (for example to get CRC status), activate the multiplex option that places M_CB[7:0] bits on DQ[7:0] by configuring the <Training Ecc Mux Enable> field in the Training Software 2 Register (Table 323 p. 666)).

To exit MPR Read mode

1. Issue an MRS command to return to normal mode.
2. Enable DLB.

9.2.2 MPR Write

To enter the MPR Write mode:

1. Configure the DRAM to MPR mode, using the DDR4_MRS3 OpCmd.
2. Configure the <MPR LOC> field and the <MPR DATA> field in the DDR4 MPR WR Register (Table 374 p. 716).
3. Perform a DDR4 MPR Wr OpCmd by configuring the OpCmd register.

The controller will issue MPR write to DRAM.

- The MPR write places write data on the A[7:0] as in the MRS command.
- It is a Write command with no DATA following it.

9.2.3 Per DRAM Accessibility (PDA)

The DDR4 protocol supports configuring specific devices using MRS commands, while masking the configuration from other devices sharing the same address and command lines.

This feature is mostly used for DDR4 Vref calibration.

To enter the PDA mode:

1. Select the octets to mask/enable by writing to the dedicated register.
2. Issue an MRS command to put the DRAM in PDA mode, using the OpCmd register.

To exit the PDA mode:

Issue an MRS command with the PDA mode disabled.

9.2.4 Unsupported DDR4 Features

The DDR4 block complies with the JEDEC DDR4 standard, and provides all the required registers. However, the SDRAM controller does not support the following DDR4 features:

- Write DBI
- Write CRC
- CAL
- CA Parity

Document Classification: Proprietary Information

10 NAND Flash Controller (NFC)

This section describes the device NAND Flash Controller (NFC).

The device includes the NAND Flash Controller Version 2 (NFCv2). The NFCv2 is used to interface with an external NAND flash. The usage model for the NAND flash is that of a Solid State Drive for the device. A flash file system (FFS) performs tasks such as wear leveling, garbage collection, and bad-block replacement for the flash device. The NFCv2 accepts commands from the file system, controls the flash, according to these commands, and performs error control.

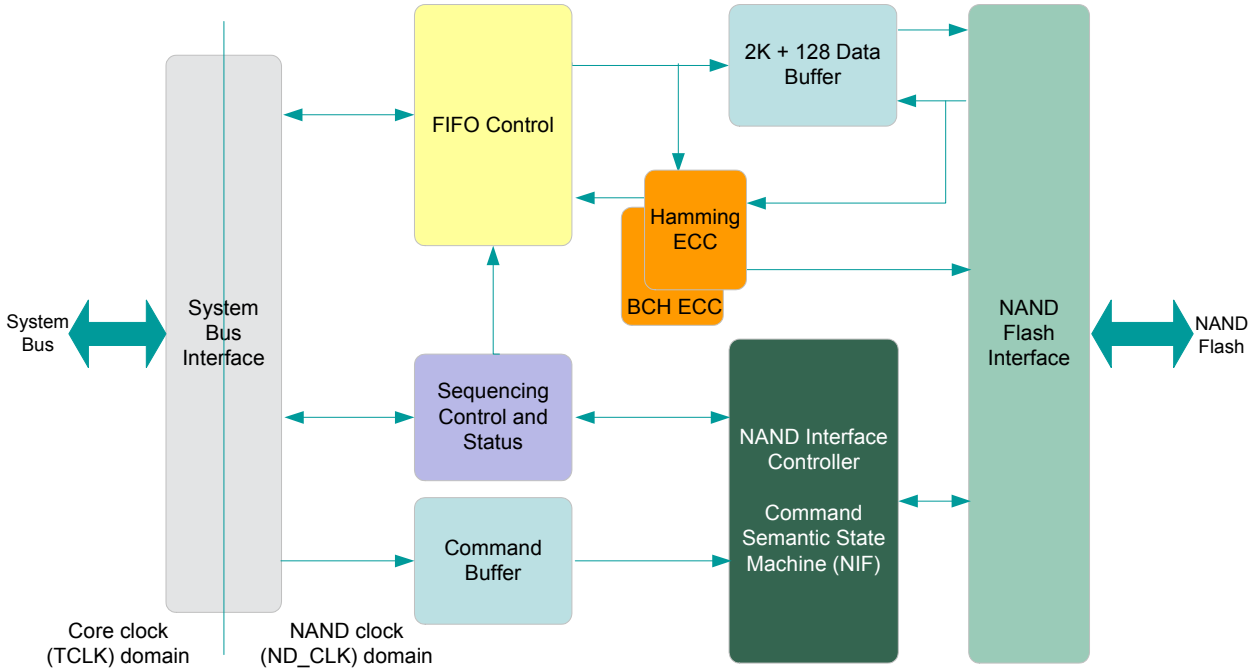
Flash devices use a multi-cycle addressing scheme, so the number of interface pins remains the same for all memory densities.

- ND_IO[7:0] are used for sending command and addresses to the flash device, in both 8-bit and 16-bit NAND flash.
- ND_IO[7:0] are also used for data transfer with 8-bit devices. ND_IO[15:0] are used for data transfer with 16-bit devices.

The NFCv2 controls these interface pins for the specific command placed in its command buffer.

The NFC registers are located in Appendix A.4, NAND Flash Registers.

Figure 13: NAND Flash Controller Block Diagram



10.1 Features

The following major features are integrated in the interface:

- Supports 4 Chip Enable's (CE) and 8-/16-bit interface to the flash device.
- Supports 32/64/128/256-KB block sizes.
- Supports 512B, 2-KB, 4-KB, and 8-KB page sizes.
- Computes Error Correction Code (ECC) and corrects single-bit errors and detects 2-bit errors per page using Hamming ECC.
- Computes ECC and corrects up to 16 errors per page (including spare, if enabled and parity bits themselves) using BCH[1] ECC.
- Supports programmable interface timing.
- Supports enabling Interrupts to indicate page and command completion, bad blocks, bit errors, flash-ready status and command, and data-write/read requests.
- Supports boot from NAND flash on ND_CEn[0].

Note The total storage that the NFCv2 can access is limited by the number of CE pins. The addressing is sufficiently flexible to enable any size NAND device, but any configuration requiring more than four chip enable signals is not directly supported by this controller.

10.1.1 NAND Attribute Definitions

Table 10 lists the key high-level attributes of NAND devices.

Table 10: NAND Attribute Definitions

Attribute	Definition
Density	The total storage capacity of the NAND device, typically in units of Mbits or Gbits, also denoted Mb or Gb. The true usable density may vary depending on file system overhead, garbage collection policy, and bad blocks. This controller supports a wide range of densities by virtue of its support for both 4-cycle and 5-cycle addresses, and by stacking NAND devices using multiple CEs.
Interface Width	Number of bits of the NAND data bus. "Raw NAND" is available in 2 widths—8 and 16 bit. This controller supports both widths.
Interface Speed	This is specified differently depending on the nature of the specific interface. For Raw NAND devices, the interface is asynchronous and the interface speed is the read/write pulse cycle time, which is specified in nanoseconds. This controller supports varying interface speeds by enabling programmability into the timing generation.
NAND Cell Type	Single Level Cell (SLC) or Multi Level Cell (MLC).
ECC	Error Correction Coding. This indicates the recommended number of bits that should be corrected by the ECC solution on the controller. Typically the controller would require the capability to detect more errors than the minimum ECC requirement. Typically ECC requirements are specified in the number of bits per 512B section, or alternatively as the number of bits per page and this requirement can vary from vendor to vendor. The ECC algorithm can be simple for single bit correction (for example, Hamming code) or complex for multiple bit correction (with manufacturers suggesting Reed-Solomon or Bose-Chaudhuri-Hochquenghem).

1. BCH = Bose-Chaudhuri-Hocquengham.

Copyright © 2015 Marvell

Table 10: NAND Attribute Definitions (Continued)

Attribute	Definition
Page Size	This is the number of bytes in a single chunk of the array that can be read or written with a single command. The page size can be considered to be just the data area (and therefore be a power of two in size), or can be the data area plus the “spare” area (therefore, cannot be a power of 2). This controller supports a range of page sizes.
“Spare” Area	This is additional bits at the end of the page typically used for overhead functions. ECC syndrome bits may be stored here or file system “meta data” may be here. The spare area is considered out-of-band storage. Generally, there are 16B of spare per 512B of page size. MLC NAND increases the needed overhead dedicated to the spare area because of the potentially extreme ECC levels needed.
Garbage Collection	NAND is written in pages (for example, 2048B) and pages are combined into erase blocks (for example, 64 pages in a block). New data can obsolete already written data where the old data has not yet been erased. As the NAND fills up, the old obsolete data needs to be garbage collected to make room. This typically involves moving the remaining “good” data in a block being garbage collected to an erased block to make all blocks in an old erase block obsolete. The old block is then erased (that is, garbage collected). The newly written block contains only valid and current data.
Endurance	**NOTE:** This is the number erase/program cycles that can reliably be supported by the device over the complete lifetime of the device. Currently, for SLC devices this is ~100,000 and for MLC devices this is ~10,000.

10.2 NAND Flash Interface

The NFCv2 supports both 8- and 16-bit-wide data buses. Up to 4 CEs (ND_CEn[3:0]) interface to the flash devices. Up to 5 cycles of addresses can be sent on the ND_IO bus to address flash devices interfaced, using these CEs. The CE to activate can be specified in the command for the NFCv2.

10.3 NAND Flash Connectivity

One significant feature present in newer NAND devices is the presence of a data cache that allows data reads and writes to be pipelined to the device. Unfortunately, manufacturers have different logic semantics on the RBn pins when using caching.

The read access time is typically relatively short in comparison to the time required to read the data from the cache buffer. Using the data cache, it is possible to perform extended transfers at essentially full transfer speed.

For Write operations, the program time is typically 10 times longer than that of a Read access. In this situation, the cache significantly increases the write throughput, but it is not possible to sustain full transfer speed to a single device. Using striping, as shown in Figure 15, across the NAND controller’s two chip enables allows for doubling the write performance, so long as the writes can be scheduled.

The device integrates 2 RBn signals and up to 4 CE signals. There is one ND_RBn signal per CE or CE pair, as configured in the NAND Flash XBAR Configuration Register (ND_CEn[0] and ND_CEn[2] share RBn[0]. ND_CEn[1] and ND_CEn[3] share RBn[1]). This additional interface ability is intended to support stacked NAND to increase the memory density.

A selection within the CE pair is made by decoding address bit[39]. To select which CE is used, set the <ADDR5> field in the NAND Controller Command Buffer 2 (NDCB2) Register (Table 301 p. 542) and the <CSEL> field in the NAND Controller Command Buffer 0 (NDCB0) Register (Table 299 p. 540) as listed in Table 11.

To activate ND_CEn[2] and ND_CEn[3], write 1 to <NfCsExpansionEn> field in the SoC Device Multiplex Register (Table 1889 p. 1396).

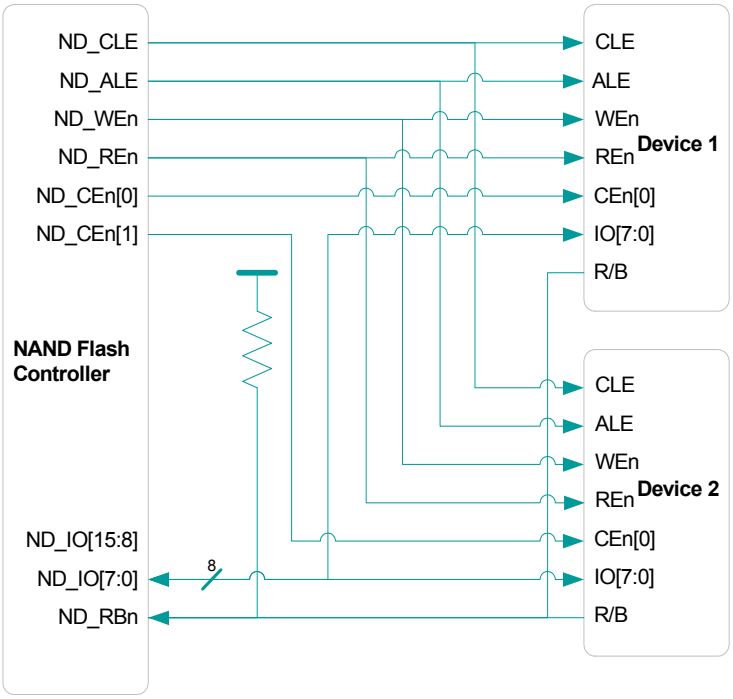
Copyright © 2015 Marvell

Table 11: CE Assignment

Selected CE	NDCB2 <ADDR5> bit[7]	NDCB0 <CSEL> bit[24]
ND_CEn[0]	0	0
ND_CEn[1]	0	1
ND_CEn[2]	1	0
ND_CEn[3]	1	1

Different types of configuration are illustrated in the following figures.

Figure 14: Stacked Flash System Example Using ND_CEn[0] and ND_CEn[1]



Copyright © 2015 Marvell

Figure 15 shows the dual CE and RBn connectivity using ND_CEn[0] and ND_CEn[1].

Figure 15: Striped Flash System Example Using ND_CEn[0] and ND_CEn[1]

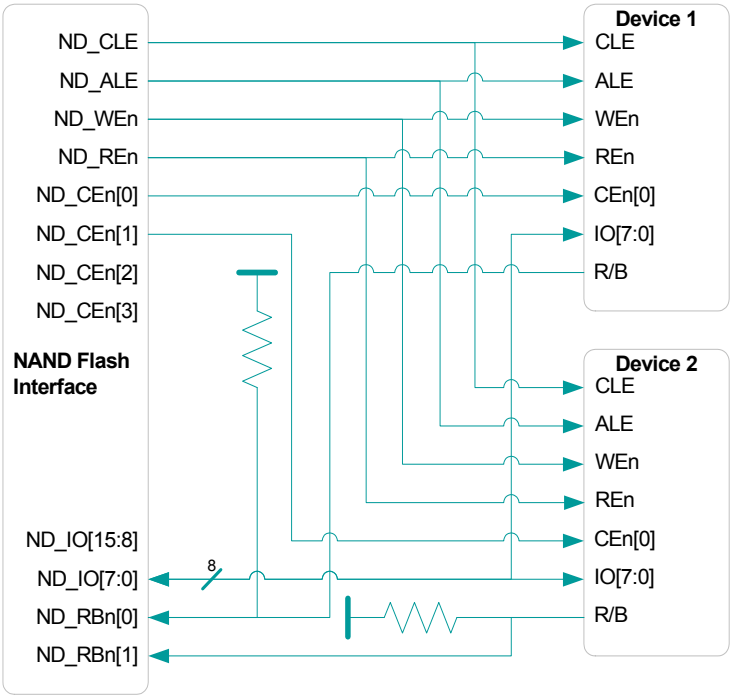
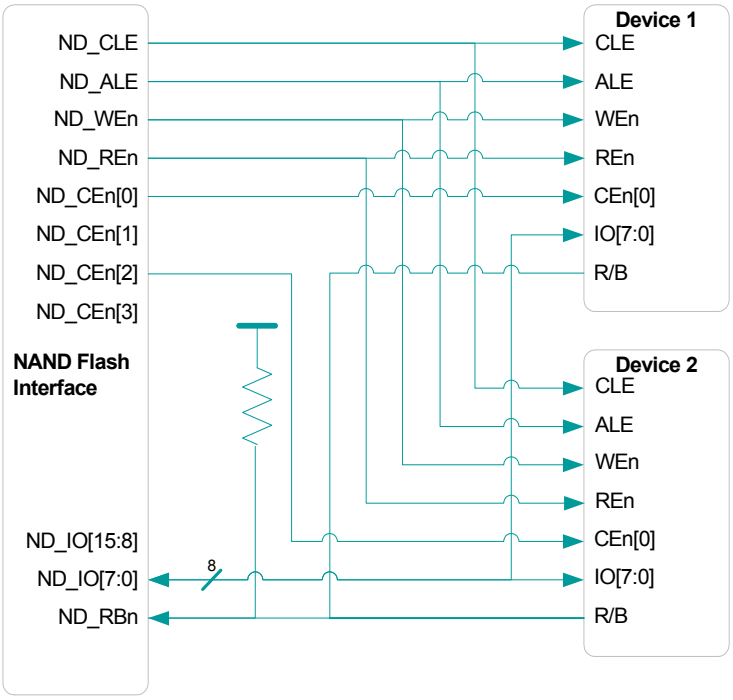


Figure 16 shows dual CE using ND_CEn[0] and ND_CEn[2].

Figure 16: Stacked Flash System Example Using ND_CEn[0] and ND_CEn[2]



Copyright © 2015 Marvell

10.4 Operation

NAND flash devices accept a variety of commands from the NFCv2 to perform functions such as program, erase, and different types of reads. This section provides information for configuring the NFCv2 to perform successful program, erase, and reads.

- Error checking and correction (ECC)
- Bad-block management support

10.4.1 Error Checking and Correction (ECC)

Error-detection code/error-correction code (EDC/ECC) is required in the NFCv2 to detect and correct errors occurring in the flash device due to bit flipping (Bit flipping occurs when a bit is either reversed, or is reported reversed). ECC is computed when write data is transferred from data buffer to the NAND flash. After completing the write, the computed ECC bytes are written to the flash. Table 12 shows the available spare area and spare area bytes used for storing the ECC for different page sizes. When ECC is enabled, the number of spare area bytes required for use by ECC are shown in the middle column of Table 12. These bytes are not available for use by the system software and cannot be written to or read from.

In addition, the number of bytes read from and written to the NAND that pass through the data FIFO must be a multiple of eight.



Note

Enable ECC only for Program page or Read page operations. Therefore, for Read-ID and Read-status operations disable ECC mode.

To interleave Read-ID with other operations, the control and the command registers have to be reconfigured for the Read-ID, and therefore, ECC setting can be disabled.

For example, if the page size is 512B with ECC and the spare area enabled, bytes 0 through 511 are written with data and the first 8B of the spare area (bytes 512 through 519) are written with spare data (file-system-dependent information). Next, 6B of spare area (bytes 520 through 525) are used by the NFCv2 to write the ECC data. The remaining 2B cannot be written and any data read must be ignored.

When the read operation is completed, ECC bytes are read from the spare area, and any bit errors are computed and corrected. Depending on the ECC mode, the NFCv2 uses Hamming code to correct 1-bit random error in a page and detects 2-bit errors or a BCH algorithm to correct multiple bit errors. The ECC Failure error mask (UNCERRM) bit is used to mask or enable an interrupt request that is asserted when an uncorrectable Hamming double-bit error is detected in any of the data streams as indicated by NDSR[UNCERR]. When in extended ECC mode, this bit indicates a failure with the BCH ECC.

When the 16-bit NFCv2 flag is set to ON, there is an interaction between the ganging configuration and the ECC option available:

- Ganging of 512B page size devices is only supported with the Hamming ECC engine
- Ganging of 2048B (or 4/8KB) page size devices is only supported with the BCH ECC engine.

Copyright © 2015 Marvell

Table 12: Spare Area Used for ECC

NAND Flash Page Size (Bytes)	Typical Spare Area in NAND Flash (Bytes)	Number of Spare Area Bytes Used for Hamming ECC	Number of Spare Area Bytes Used for BCH ECC	Number of Inaccessible Bytes
512	16	0	N/A	0
512	16	6	N/A	2
2048	64	0	0	0
2048	64	24	0	0
2048	64	0	30 **NOTE**: For NAND with up to 4 bit ECC	2
4096	128	48		0
4096	128	0	60 **NOTE**: For NAND with up to 4 bit ECC	4

10.4.2 Hamming Code for ECC

During a page write, data is split into even and odd streams before computing the ECC. This data split lessens the error-correcting requirement on the ECC algorithm to 1 bit. If D_{ij} represents the j^{th} bit of the i^{th} byte, Table 13 and Table 14 show the data streams used for ECC computation for an 8-bit wide data bus. When the 16-bit NFCv2 flag is set to ON, for NAND flash devices that are 16-bits wide, the lower and upper bytes are treated as successive bytes of read data. However, where two 8-bit devices are interfaced through a single CE, the ECC computations are performed independently on the upper and lower bytes of the data bus. Figure 17 illustrates the computation of partial parities (indicated by Px for odd numbered bits and Px' for even numbered bits) for a 256B data stream. ECC is computed separately for each data stream and is written to the spare area of the flash. The ECC algorithm can correct 1 bit of random error in a chunk of 256B, de-interleaved from 512B. Each of the odd and even data streams are 256B long, and the computed ECC for this data stream occupies 3B. Therefore, 6B of ECC data are written for a NAND flash with page size of 512B. These bytes are stored into the spare byte location in the NAND flash memory, as indicated in Table 12, using the same notation for partial parity bits as in Figure 17. The same ECC computation process is employed 4 times for a page size of 2048B, resulting in 24B of ECC.



Note

Hamming ECC is not supported with the 4-KB (or larger) logical page size. Therefore, for such logical pages, use BCH ECC.

Copyright © 2015 Marvell

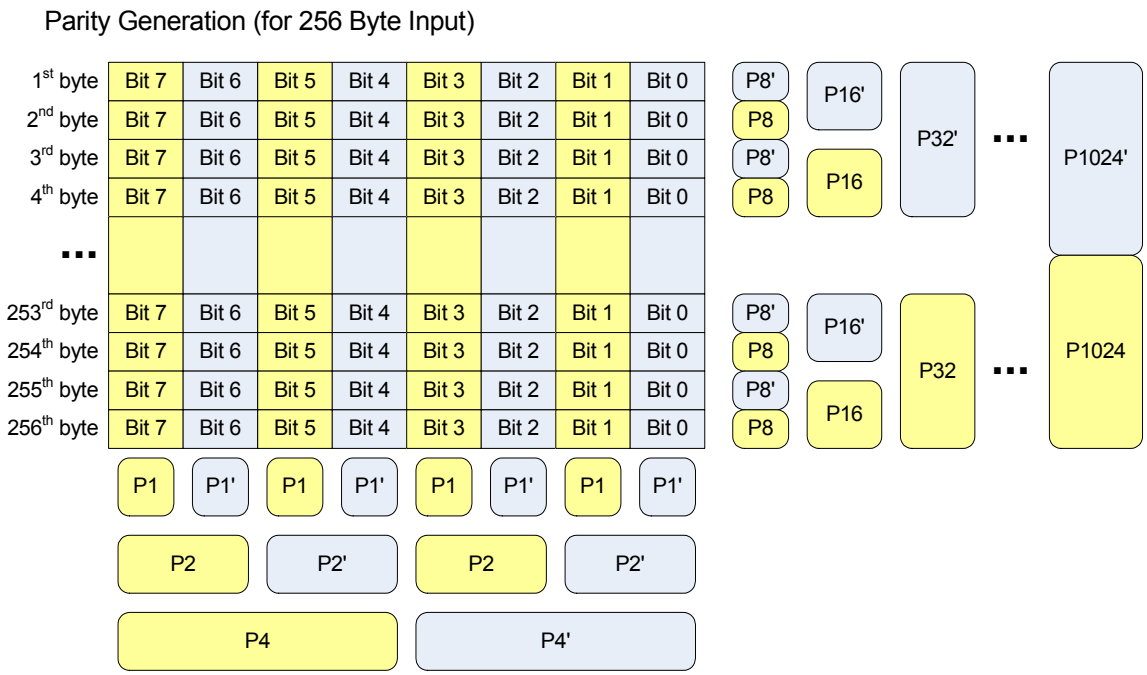
Table 13: Even Data Stream

Bit	7	6	5	4	3	2	1	0
	$D_{i+1\ 6}$	$D_{i+1\ 4}$	$D_{i+1\ 2}$	$D_{i+1\ 0}$	$D_{i\ 6}$	$D_{i\ 4}$	$D_{i\ 2}$	$D_{i\ 0}$

Table 14: Odd Data Stream

Bit	7	6	5	4	3	2	1	0
	$D_{i+1\ 7}$	$D_{i+1\ 5}$	$D_{i+1\ 3}$	$D_{i+1\ 1}$	$D_{i\ 7}$	$D_{i\ 5}$	$D_{i\ 3}$	$D_{i\ 1}$

Figure 17: ECC Code Generation



In Table 15, *p* refers to the partial parity bit for the first 256B and *P* is the partial parity for the second 256B. The “first” and “second” are not consecutive by address, but interleaved by bit:

- The “first” 256 being composed of the even bits.
- The “second” 256 being composed of the odd bits.

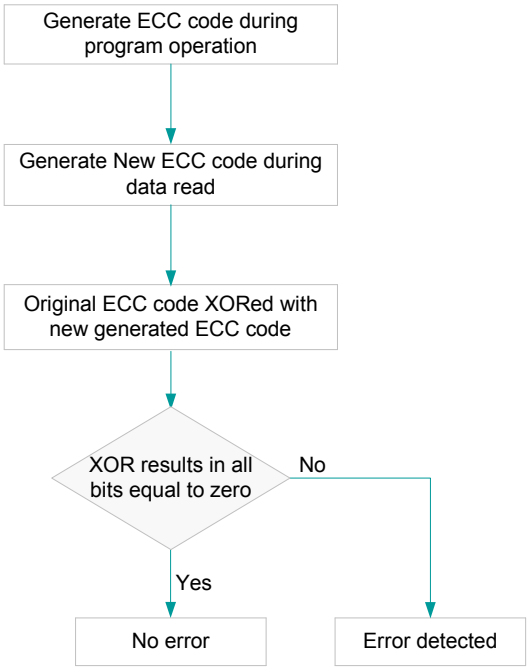
Table 15: ECC Byte Placement

Bit	7	6	5	4	3	2	1	0
Byte 0	P8'	p8'	p4'	p4'	p2'	p2'	p1'	p1'
Byte 1	p128'	p128'	p64'	p64'	p32'	p32'	p16'	p16'
Byte 2	P1	P1	p1024'	p1024'	p512'	p512'	p256'	p256'
Byte 3	P16	P16	P8	P8	P4	P4	P2	P2
Byte 4	P256	P256	P128	P128	P64	P64	P32	P32
Byte 5	0	0	0	0	P1024	P1024	P512	P512

During a page read, the read data is again split into odd and even streams. The associated ECC data is read and compared with the computed ECC from the received data stream. If a 1-bit error is detected in a data stream, it is corrected. For two errors in any or both of the data streams, the ECC algorithm detects this scenario and flags a 2-bit error condition.

The ECC logic is initialized at the end of every 512B (256B per data stream) of transfer.

Figure 18: Hamming Error Detection Process



Copyright © 2015 Marvell

10.4.3 BCH Error Correction

BCH is an acronym for Bose, Ray-Chaudhuri, Hocquengham, three individuals who collectively developed the mathematical background for multilevel, cyclic, error-correcting, variable-length digital codes. There are three parameters which define BCH codes: symbol size, field length and correction power.

For this purpose, the symbol is a binary digit. The NAND flash interface is byte parallel whereas the BCH algorithm is inherently bit serial. The mathematics of BCH is based on operations on Galois fields where the size of the field is (2^m -1) symbols (bits). Since BCH error correction is not intended for small block NAND flash devices, the field is chosen to cover all 2048B in a large block page, plus any spare bytes that are defined.

There are 16,384 bits in 2048 bytes, but when you add in the ECC bits and the spare bits (which are also protected), the next larger power of two (minus one) is required that results in 32,767 bits. Therefore, the BCH parameter for the field size “m” is 15. The inherent capability of a BCH algorithm means that the ECC corrects any combination of n errors in 2^m -1 bits.

The NAND flash manufacturers, for reasons dating back to a small block NAND flash, specified the ECC requirements for MLC as the number-of-bits-of-correction per 512B sector. The statistics for the bit error rate (BER)[1] for each NAND flash cell for MLC is uniform and the division of those statistics into 512B sectors is arbitrary. For BCH, the number of correctable bits is the parameter ‘t’.

For example, if the requirement is for 8 bits of correction per 512B sector, the implication is that there is a requirement of 32-bits per 2-KB page. However, there is no way to prevent 9 of those bits from being in the first sector, or 10, or 11, or any combination up to the maximum of 32, it is solely a function of the probabilities and possible combinations.

After a massive combinational analysis where all possibilities of 32 bits of errors distributed over four 512B sectors, it was discovered that fewer total bits of correction, when designed to cover the entire page, give better results than dividing the page into arbitrary chunks. It turns out that 16-bits of correction per page gives better coding gain (that is, an increase in effective BER) than four 8-bit, 512B sector correction.

The trade-off is that the amount of hardware is approximately proportional to (m*t) so increasing t and m simultaneously (more bits of correction over a larger field) significantly increases the number of gates. However, the BCH algorithm is highly sensitive to the particular values of m and t, and all of the polynomials used are chosen specifically for each choice. This means that a generic or programmable BCH engine is not a practical alternative so a choice has to be made as to how far into the future one wishes to be able to intercept MLC BER rates.

10.4.4 Page Allocation

There are three potential types of data in a page, the main data, the so-called spare data, and the ECC. In addition, in certain common cases, some bytes are simply inaccessible. The layout of this data depends on the programming of <PAGE_SZ>, <SPARE_EN>, and <ECC_EN> in the Data Flash Control Register (NDCR) (Table 290 p. 518), and <BCH_EN> in the NAND ECC Control (NDECCCTRL) Register (Table 296 p. 535). The following table shows how bytes are allocated in the page, according to the various page sizes. Because of the 64-bit data buffer, the size of any area must be evenly divisible by 8B, so certain combinations result in unusable spare bytes. Unusable bytes must be programmed to zero in data blocks.

For small block devices, there can be a single Hamming 6B ECC, and its location depends on the spare usage. For small block devices, there is a maximum of 16B of spare area. Since the BCH ECC produces a 30B syndrome, BCH cannot be used for small block NAND flash.

1. The term BER is defined as the probability of error in any one bit.

Copyright © 2015 Marvell

Table 16: Small Block Page Allocation

PAGE_SZ == 0b00 (512 bytes)	SPARE_EN == FALSE	ECC_EN == FALSE	BCH_EN == don't care
0 … 511		512 … 527	
512B Data		Inaccessible	

PAGE_SZ == 0b00 (512 bytes)	SPARE_EN == TRUE	ECC_EN == FALSE	BCH_EN == don't care
0 … 511		512 … 527	
512B Data		16B Spare	

PAGE_SZ == 0b00 (512 bytes)	SPARE_EN == FALSE	ECC_EN == TRUE	BCH_EN == FALSE
0 … 511		512 … 517	518 … 527
512B Data		Hamming ECC	Inaccessible

PAGE_SZ == 0b00 (512 bytes)	SPARE_EN == TRUE	ECC_EN == TRUE	BCH_EN == FALSE
0 … 511		512 … 519	520 … 525	526, 527
512B Data		8B Unprotected Spare	Hamming ECC	Inaccessible

For large block devices, there can be 4 concatenated 6B Hamming ECC syndromes or a single 30B BCH syndrome (in case of NAND device with up to 4 bit ECC). The location of the ECC depends on the spare usage.

Table 17: Large Block Page Allocation

PAGE_SZ == 0b01 (2048B)	SPARE_EN == FALSE	ECC_EN == FALSE	BCH_EN == don't care
0 … 2047		2048 … 2111	
2048B Data		Inaccessible	

PAGE_SZ == 0b01 (2048B)	SPARE_EN == TRUE	ECC_EN == FALSE	BCH_EN == don't care
0 … 2047		2048 … 2111	
2048B Data		64B Spare Data	

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 17: Large Block Page Allocation (Continued)

PAGE_SZ == 0b01 (2048B)	SPARE_EN == FALSE	ECC_EN == TRUE	BCH_EN == FALSE
0 – 2047		2048 – 2071	2072 – 2111
2048B Data		(4) Hamming ECC	Inaccessible

PAGE_SZ == 0b01 (2048B)	SPARE_EN == TRUE	ECC_EN == TRUE	BCH_EN == FALSE
0 – 2047		2048 – 2087	2088 – 2111
2048B Data		40B Spare Data	(4) Hamming ECC

PAGE_SZ == 0b01 (2048B)	SPARE_EN == FALSE	ECC_EN == TRUE	BCH_EN == TRUE
0 – 2047		2048 – 2077	2078 – 2111
2048B Data		BCH ECC	Inaccessible

PAGE_SZ == 0b01 (2048B)	SPARE_EN == TRUE	ECC_EN == TRUE	BCH_EN == TRUE
0 – 2047		2048 – 2079	2080 – 2109, 2110, 2111
2048B Data		32B Spare Data	BCH ECC, Inaccessible

It is expected that all NAND flash devices require ECC, so all examples include some type of ECC.

Note

For 4KB pages of type SLC or MLC with 4-bit ECC, always use 2KB chunks packed together in some manner. The same applies to an 8KB page.

For the BCH ECC examples, the BCH ECC must be padded with two zero bytes to maintain the 8B alignment.

If 4KB pages scale, there would be 4096B of main area and a spare area of 128B. It is expected that there will be more than 128B of spare area (due to MLC requirements), but these additional bytes will not be accessible by the NFC.

Copyright © 2015 Marvell

Table 18: 4-KB Physical or Logical Page Allocation Using 2-KB Chunks

PAGE_SZ == 0b01 (2048B)					SPARE_EN == FALSE				ECC_EN == TRUE				BCH_EN == FALSE		
0		2047	2048		2071	2072		4119	4120		4143	4144...			
2KB Data			(4) 6B Hamming ECC			2KB Data			(4) 6B Hamming ECC			Inaccessible			
PAGE_SZ == 0b01 (2048B)					**SPARE_EN == FALSE**				**ECC_EN == TRUE**				**BCH_EN == TRUE**		
0		2047	2048		2077	2078		4125	4126		4155	4156...			
2KB Data			30B BCH ECC			2KB Data			30B BCH ECC			Inaccessible			
PAGE_SZ == 0b01 (2048B)					**SPARE_EN == TRUE**				**ECC_EN == TRUE**				**BCH_EN == FALSE**		
0		2047	2048, 2087	2088, 2111		2112		4159	4160, 4199	4200, 4223		(4224...)			
2KB Data			40B Spare	(4) 6B Hamming ECC		2KB Data			40B Spare	(4) 6B Hamming ECC		Inaccessible			
PAGE_SZ == 0b01 (2048B)					**SPARE_EN == TRUE**				**ECC_EN == TRUE**				**BCH_EN == TRUE**		
0		2047	2048 - 2079	2080 - 2109		2110		4157	4158 - 4189	4190 - 4219		(4220...)			
2KB Data			32B Spare	30B BCH ECC		2KB Data			32B Spare	30B BCH ECC		Inaccessible			

The two ECC cases provide two different strengths for ECC. The Hamming ECC is intended for (but not exclusive to) SLC NAND flash. The SLC NAND flash specification currently specifies that there should be 1 bit of correction per 512B sector. This device provides 2-bit detection per 512B sector, so the ECC goes beyond the requirements of the specification.

BCH ECC is intended for (but not exclusive to) MLC NAND flash. The MLC NAND flash specification currently specifies 4 bits of correction per 512B sector, with the next generation's requirements increasing to 8 bits of correction. The BCH engine implements 16-bit correction across the entire 2-KB chunk, and because of the increased combination coverage of possible errors, this provides several orders of magnitude greater BER coding gain than the 8-bit sector approach.

For MLC NAND with stronger ECC requirements, a reduction in the chunk size from 2048 to 1024 (or any other size, which is a multiple of 32B) can be made using the variable length register. This will reduce the field size over which the 16-bits of correction is applied, although because of the lower combination coverage, it will not have the same strength as a full page correction.

Technically this "wastes" 2B of BCH syndrome, because it would be possible for a 1024B field to only require a 28B syndrome, but this approach provides nearly double the coverage without redesigning the controller.

Document Classification: Proprietary Information

The drawback in this case is that essentially the entire currently defined spare area would be consumed with ECC bytes, and for each 2-KB page, there would be four “wasted” bytes.

This can be extended yet again by using BCH on a 512B sector, again wasting 4B per syndrome. For this to operate, there would need to be a doubling of the spare area from 64B per 2-KB page to at least 120B per 2-KB page.

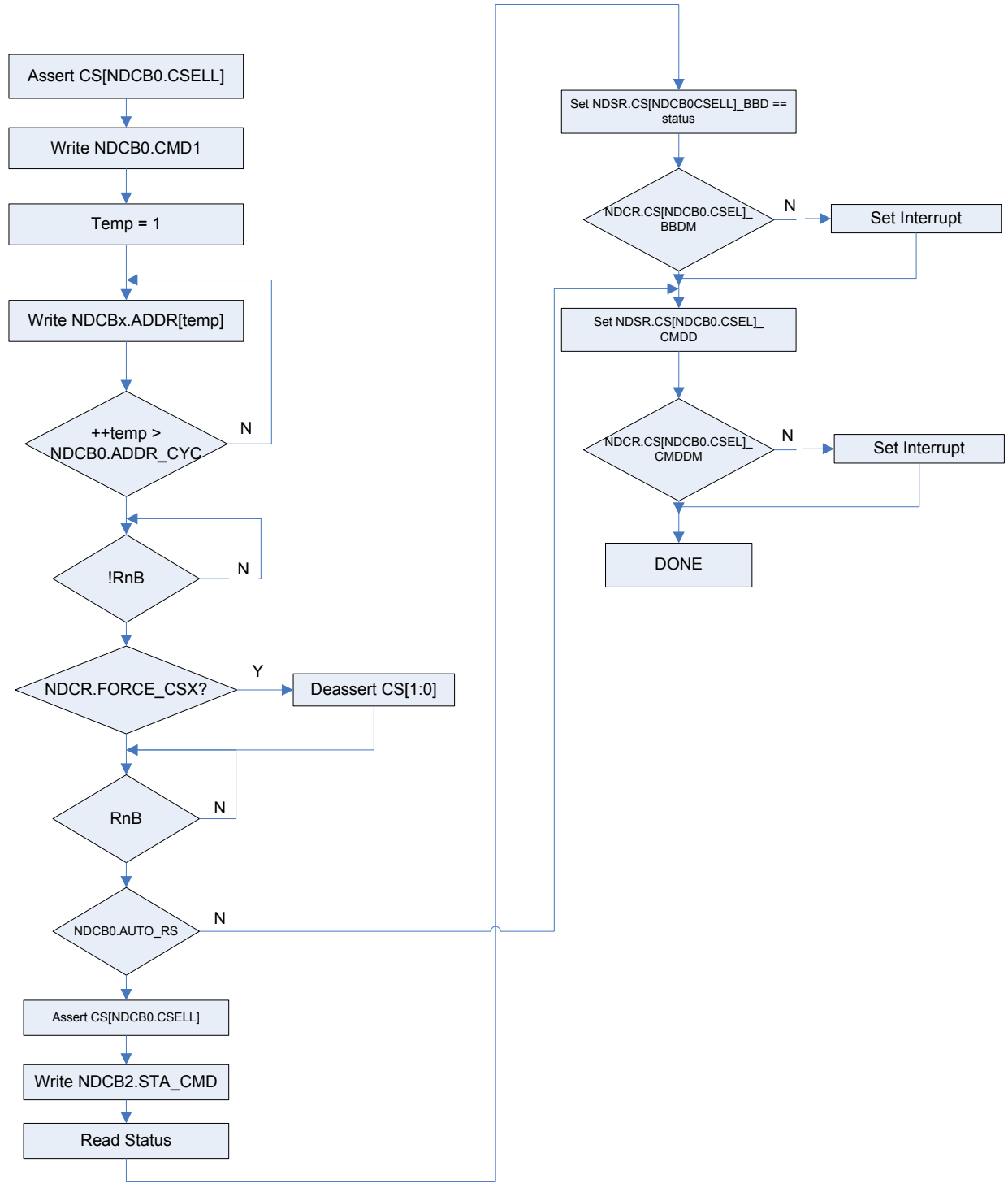
The same logic applies to 4-KB pages.

10.4.5 Command Semantics

The command semantics are defined in the NAND Controller Command Buffer 0 (NDCB0) Register (Table 299 p. 538). This register defines how each NAND command type is to proceed, and how or if it may be overlapped. There are eight basic semantics: Read, Program, Erase, Read ID, Read Status and Reset, Naked Command and Naked Address. The semantics differ in the sequence of operations and in the count of operations. However, each semantic runs as a continuous operation. Read and Program semantics can be decomposed into their “naked” segments, and Program and Erase semantics allow overlap of commands to a different CE.

Copyright © 2015 Marvell

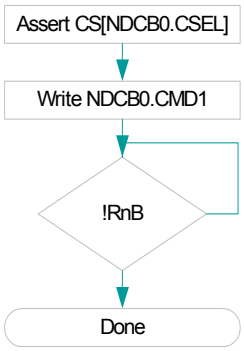
Figure 19: Erase Semantic Flowchart



Copyright © 2015 Marvell

- **Read ID Semantic** The Read ID semantic is very similar to the Read semantic, only the length is fixed by a a different field to be either 2 or 4B. There is no concept of spare or ECC, but the flowchart is very similar, and therefore, is not replicated.
- **Read Status Semantic** The Read Status semantic is very similar to the Read semantic, only the length is fixed to 1B. There is no concept of spare or ECC, but the flowchart is very similar and therefore, is not replicated.
- **Reset Semantic** The Reset semantic issues a single command (CMD1) to the NAND, regardless of the programming of other command block parameters (such as NDCB0.DBC). Address bytes have no meaning, regardless of the address count. There is no data transfer regardless of any length settings.

Figure 20: Reset Semantic Flowchart



10.4.6 Bad Block Management Support

The NFCv2 performs a status check after each program/erase operation to determine if the transaction was successful. This status check can be disabled by clearing the NAND Controller Command Buffer 0 (NDCB0) Register (Table 299 p. 538). If a status check returns an error, as indicated by a 1 in the LSB of the status-read data, a bad-block-detect interrupt (if enabled) is sent out. The Bad Block registers (see Table A.4, NAND Flash Registers, on page 517) can be read to determine the address of the bad block. The OS file system software marks the detected bad block as non-valid.

- If a bad-block error is encountered at the end of a page program, additional transactions to that block are aborted.
- If the error occurs during a page program in block A, software transfers other valid pages in block A to another unassigned block B and marks block A as non-valid.
- If an error occurs during a block erase, software is expected to mark the block as non-valid.

Command Execution When Bad Blocks are Detected

When a bad block is detected, the behavior of the NFCv2 remains the same, regardless of whether commands are executed sequentially or in parallel. The command without a bad-block detection completes, but the command that resulted in bad-block detection does not execute. (If it is a program command, it must be executed again after the command descriptors have been reprogrammed to process the bad-block scenario).

Copyright © 2015 Marvell

As a result of a bad-block detection, the ND_RUN bit in NDCR is cleared after the command buffer is emptied. Resume the operation by setting ND_RUN after modifying the command-descriptor.

10.4.7 Usage Examples

The following tables show how the application processor interacts with the NFCv2 to schedule operations. These tables only document the use model. Each table shows resources in columns and steps in rows and the tables are intended to show how the hardware resources are used in sequence and in time. The right most column is the time equation (that is, how to compute the time taken for this step).

For each of the tables Table 19 through Table 22, time increases as one moves down the rows, although the time taken for each row is not constant. In most instances, the time equation column defines the time taken by each step. Each column describes the activities executed by the various hardware segments that make up the system.

Table 19 shows the sequence for reading a 2-KB page from a MLC device with BCH enabled.

Table 19: Read 2 KB MLC Page

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Decoder	Time Equation
Read command control block from memory					
Read commands from memory	Write “NAND Read” command sequence to NDCB0				
Read Data Move control block from memory		Read cmd 1			tWC
Stall		5-cycle address			5 * tWC
		Read cmd 2			tWC
	Busy				tR
				Reset	
		Read data	Write NAND flash data into Buffer	Compute syndrome	2048*tRC
		Read spare	Write NAND flash Spare into Buffer		(64-ECC)*tRC
		Read ECC			ECC*tRC
				Load BMA	
				Compute BMA	
				Load Chien search	

Copyright © 2015 Marvell

Table 19: Read 2 KB MLC Page (Continued)

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Decoder	Time Equation
Read data block from memory	Read data block from NDDB		Read buffer data to system bus	Compute ECC bits with Chien search	
	Interrupt				

Table 20 shows the sequence for reading a 4-KB page from a MLC device with BCH enabled. Because the buffer is not large enough to contain the entire page, the data is read from the NAND flash in 2-KB chunks.

Table 20: Read 4 KB MLC Page

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Decoder	Time Equation
Read Command control block from memory					
Read commands from memory	Write “NAND Read” command sequence to NDCB0				
Read Data Move control block from memory		Read cmd 1			tWC
Stall		5-cycle address			5 * tWC
		Read cmd 2			tWC
	Busy				tR
				Reset	
		Read data	Write NAND flash data into Buffer	Compute syndrome	2048*tRC
		Read spare	Write NAND flash Spare into Buffer		(64-ECC)*tRC
		Read ECC			ECC*tRC
				Load BMA	
				Compute BMA	
				Load Chien search	
Read data block from memory	Read data block from NDDB		Read buffer data to system bus	Compute ECC bits with Chien search	

Table 20: Read 4 KB MLC Page (Continued)

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Decoder	Time Equation
Read Command control block from memory					
Read commands from memory	Write “Naked Read” command sequence to NDCB0				
Read Data Move control block from memory					
Stall				Reset	
		Read data	Write NAND flash data into Buffer	Compute syndrome	2048*tRC
		Read spare	Write NAND flash Spare into Buffer		(64-ECC)*tRC
		Read ECC			ECC*tRC
				Load BMA	
				Compute BMA	
				Load Chien search	
Read data block from memory	Read data block from NDDB		Read data block	Compute ECC bits with Chien search	
	Interrupt				

Table 21 shows the sequence for writing a 2 KB page to a MLC device with BCH enabled.

Table 21: Program (Write) 2 KB MLC Page

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Encoder	Time Equation
Read Command control block from memory					
Read commands from memory	Write “NAND Write” command sequence to NDCB0				

Copyright © 2015 Marvell

Table 21: Program (Write) 2 KB MLC Page (Continued)

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Encoder	Time Equation
Read Data Move control block from memory					
Read data block from memory	Write data block to NDDB		Write system bus data to Buffer		
		Write cmd 1			tWC
		5-cycle address		Reset	5 * tWC
		Write data	Read data from buffer to NAND flash	Compute	2048*tWC
		Write spare	Read Spare from buffer to NAND flash	Compute	(64-ECC)*tWC
		Write ECC		Read Stall	ECC*tWC
		Write cmd 2			tWC
	Busy				tPROG
	Interrupt				

Table 22 shows the sequence for writing a 4 KB page to a MLC device with BCH enabled.

Table 22: Program (Write) 4 KB MLC Page

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Encoder	Time Equation
Read Command control block from memory					
Read commands from memory	Write “NAND Write dispatch” command sequence to NDCB0				
		Write cmd 1			tWC
		5-cycle address			5 * tWC
Read Command control block from memory					
Read commands from memory	Write “Naked Write” command sequence to NDCB0				

Table 22: Program (Write) 4 KB MLC Page (Continued)

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Encoder	Time Equation
Read Data Move control block from memory					
Read data block from memory	Write 2KB data block to NDDB		Write system bus data to Buffer		
				Reset	
		Write data	Read data from buffer to NAND flash	Compute	2048*tWC
		Write spare	Read spare data from buffer to NAND flash	Compute	(64-ECC)*tWC
		Write ECC		Read Stall	ECC*tWC
Read Command control block from memory					
Read commands from memory	Write “Naked Write” command sequence to NDCB0				
Read Data Move control block from memory					
Read data block from memory	Write 2KB data block to NDDB		Write system bus data to Buffer		
				Reset	
		Write data	Read data from buffer to NAND flash	Compute	2048*tWC
		Write spare	Read spare data from buffer to NAND flash	Compute	(64-ECC)*tWC
		Write ECC		Read Stall	ECC*tWC
Read Command control block from memory					
Read commands from memory	Write “NAND Write dispatch” command sequence to NDCB0				

Copyright © 2015 Marvell

Table 22: Program (Write) 4 KB MLC Page (Continued)

Application Processor and System Bus	NAND Flash Registers and Control	NAND Flash I/F Pins	Data Buffer	BCH Encoder	Time Equation
		Write cmd 2			tWC
	Busy				tPROG
	Interrupt				

10.5 Usage Models

With the introduction of the new semantics and the so-called “naked” semantics, the use of these command sequences to achieve any given goal may not be obvious. This section presents each basic command in sequence and shows how it may be used to construct the range of command sequences each command type can support.

10.5.1 NFC Timing Configuration

The NFC provides a glue-less interface to an external flash (SLC or MLC). The main NFC clock is ND_CLK. ECC is calculated according to the ECC_CLK (2 x ND_CLK).

For normal operation, ND_CLK needs to be set to 200 MHz (5nsec period).

When booting from NFC, the ND_CLK is set to 125 MHz.

These clocks are set by configuring:

- <corepll_clkdiv_nand_ecc_clk_ratio> field in the Core Divider Clock Control 1 Register (Table 1867 p. 1385)
- <core_clkdiv_reload_force> field in the Core Divider Clock Control 0 Register (Table 1866 p. 1385)
- <corepll_clkdiv_nand_ecc_clk_ratio> field in the Core Divider Clock Control 1 Register (Table 1867 p. 1385)

10.5.2 Programming the Timing Registers

The two timing registers allow for considerable flexibility in control over interface timing. NAND flash device datasheets contain a large number of timing parameters, but not all of them are equally important.

The contents of the fields for each parameter are the number of NFCv2 clocks (plus one). The granularity of the NFCv2 clock is in ND_CLK periods (denote as t(ND_CLK)). This means that all timing parameters are in increments of t(ND_CLK).

10.5.2.1 tWP, tWH (and tWC)

NAND flash datasheets have 3 components to any write cycle:

- The total cycle time (tWC)
- The active portion of the write (tWP, when ND_WEn is asserted)
- The hold portion of the write (tWH, when ND_WEn is not asserted and holding for the next tWP)

The value of tWC is not necessarily the sum of tWP and tWH as it is possible that tWC > tWP+tWH. When programming these two parameters make sure that (NDTR0CS0.tWP_NFC+1) * t(ND_CLK) is greater than or equal to the tWP parameter in the NAND flash datasheet. The same holds true for tWH. But in addition, it must to be verified that (NDTR0CS0.tWP_NFC+NDTR0CS0.tWH_NFC+2) * t(ND_CLK) is greater than the NAND flash datasheet value for tWC. It is entirely possible that NDTR0CS0.tWP_NFC and/or NDTR0CS0.tWH_NFC will need to be artificially increased to meet

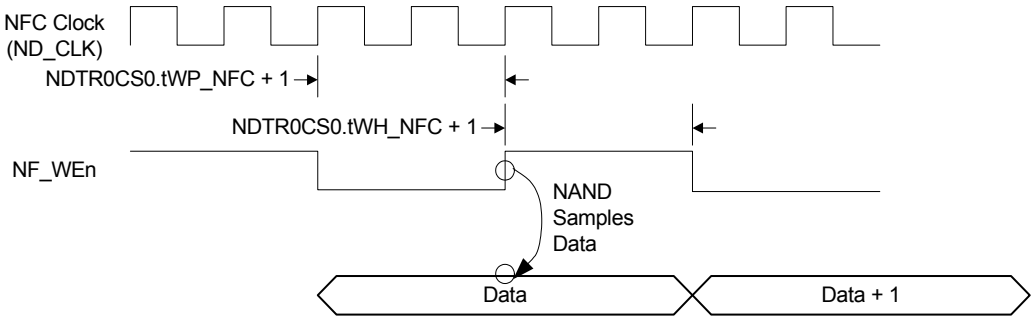
Copyright © 2015 Marvell

tWC, and if this is the case, refer to the AC timing in the device Hardware Specifications to determine the preferred parameter. If the output data valid time is critically aligned to the rising edge of ND_WEn, increase tWP_NFC, and if the data hold time to the NAND flash is critical, increase tWH_NFC.

It should be noted that these parameters effect the timing between ND_WEn edges of all write cycles of the same class. For example, a write cycle is composed of a command cycle, 5 address cycles, {page_size} data cycles and a command cycle. These parameters effect the relative write timings of the address cycles and data cycles, but the transition between command, address, and data may be effected by other timing parameters.

Figure 21 provides a timing example for these parameters. For the purpose of illustration, both tWP_NFC and tWH_NFC are set to 1, meaning that each portion of the pulse is two NFC clocks in length.

Figure 21: tWC Timing Example



The NFC clock (ND_CLK) is the internal clock for the NFCv2 controller. ND_WEn and Data are also internally generated relative to the rising edge of NFC clock. There is I/O delay between the internal representations of ND_WEn and Data due to propagation through the I/O cell and capacitive loading on the wires. However, since all signals are unidirectional (that is, being sent to the NAND flash device only), the only AC timing considerations that require analysis is the potential for a differential skew between ND_WEn and the data (which may occur because of the possibility of different capacitive loads on the two signals). This differential skew is only important if tWP and/or tWH are critically timed, according to the AC parameters in the device Hardware Specifications and the NAND flash datasheet.

Note

When the BCH ECC is used, NDTR0CS0.tWP_NFC + NDTR0CS0.tWH_NFC + 2 must be greater than or equal to 4 NFC clock (ND_CLK) cycles.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

10.5.2.2 tRP, tRH (and tRC)

NAND flash datasheets provide the following components for any read cycle:

- The total cycle time (tRC)
- The active portion of the read (tRP, when ND_REn is asserted)
- The hold portion of the read (tRH, when ND_REn is not asserted and holding for the next tRP).

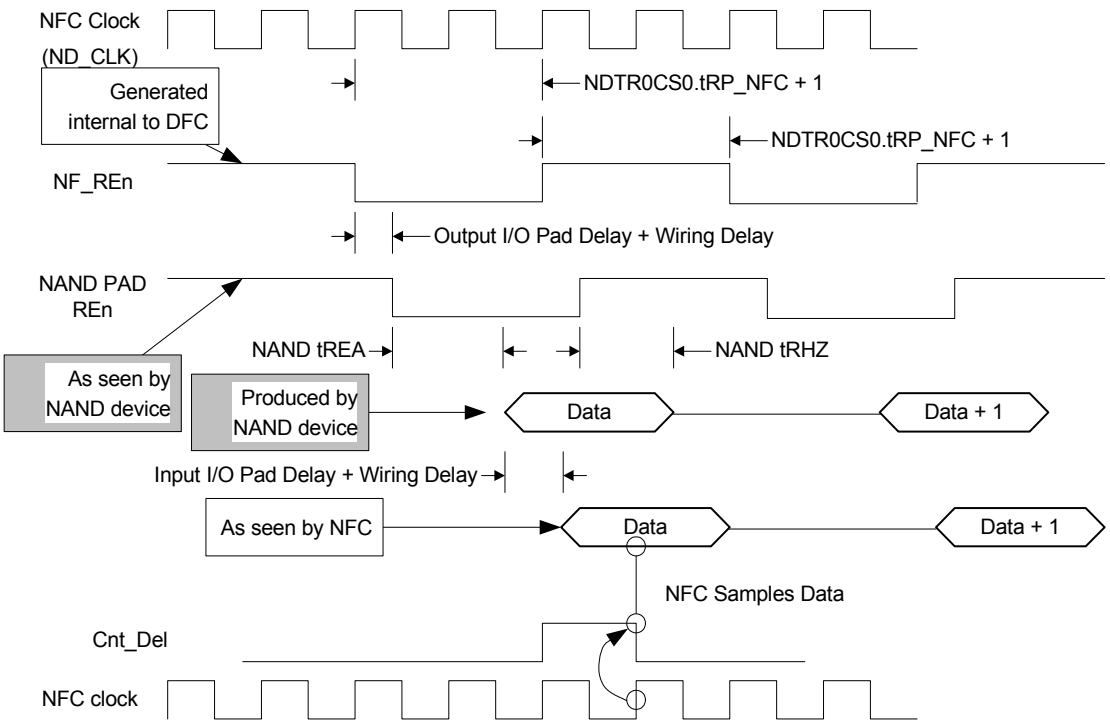
The value of tRC is not necessarily the sum of tRP and tRH, as it is possible that tRC > tRP+tRH. When programming these two parameters make sure that (NDTR0CS0.tRP_NFC+1) * t(ND_CLK) is greater than or equal to the tRP parameter in the NAND flash datasheet. The same holds true for tRH. But in addition, it must be verified that (NDTR0CS0.tRP_NFC+NDTR0CS0.tRH_NFC+2) * t(ND_CLK) is greater than the NAND flash datasheet value for tRC. It is entirely possible that NDTR0CS0.tRP_NFC or NDTR0CS0.tRH_NFC will need to be artificially increased to meet tRC. If this is the case, refer to the AC timing in the device Hardware Specifications to determine the preferred parameter. If the access time from the NAND flash is critical, increase the NDTR0CS0.tRP_NFC value, and if the hold time to the NFCv2 is critical, increase the NDTR0CS0.tRH_NFC value. In addition, there is skew between the internal NFC clock and I/O pad that drives ND_REn. This skew effectively adds to tRP because ND_REn is “seen” by the NAND flash chip “late”. In addition, data presented to the NFCv2 by the NAND chip has to go through I/O buffers, and this increases both the access and the hold time.

If NDTR0CS0.Rd_Cnt_Del is selected by NDTR0CS0.SelCntr, the sample clock is a fixed integer count from the rising edge of the NFC clock (ND_CLK) that generates the falling edge of ND_REn. Figure 22 shows this alternative sample mechanism. In this figure, it is assumed that the I/O delays from the NFCv2 to (and from the NAND flash device) place the Data window one full NFC clock cycle after the rising edge of ND_REn. This requires that NDTR0CS0.Rd_Cnt_Del have the value of 2.

Copyright © 2015 Marvell

In Figure 22, the NFC clock is shown twice only to simplify the figure.

Figure 22: tRC Timing Example Using Cnt_Del



Note

- When the BCH ECC is used, NDTR0CS0.tRP_NFC + NDTR0CS0.tRH_NFC + 2 must be greater than or equal to 4 NFC clock (ND_CLK) cycles.
- Since the NOR/NAND hardware arbiter is always enabled by default, the software must add a value of 3 to the <Rd_Cnt_Del> field in the NAND Interface Timing Parameter 0 Register (NDTR0CS0) (Table 291 p. 525) to compensate on internal sampling stages.

10.5.2.3 tCS

The NDTR0CS0.tCS_NFC parameter is used in four distinct contexts, only one of which is to address the NAND flash tCS parameter. The tCS parameter on NAND flash datasheets specifies the time between the assertion of CE and the rising edge of ND_WEn. For the NFCv2, the specification of tWP specifies a portion of the tCS, because ND_WEn is asserted at a different time than ND_CEn. For this reason, the NDTR0CS0.tCS_NFC parameter actually specifies the positive difference between the value specified by NDTR0CS0.tWP and the value needed for tCS.

For example, where the NAND device tWP specification is 15 ns and the tCS specification is 30 ns, and the NFC clock period (t(ND_CLK)) is 5 ns. NDTR0CS0.tWP must be set to the value 2 so that (2+1) * 5 = 15 ns, which meets the tWP NAND flash specification. NDTR0CS0.tCS must be set to a value that makes up the difference between 15 and 30. Since this means an additional 15 ns is required, and this value is obtained by setting tCS to the value of 2 so that (2 + 1) * 5 = 15 ns. The NAND flash requirement (15) is met in this case, since 15 + 15 = 30 ns.

Copyright © 2015 Marvell

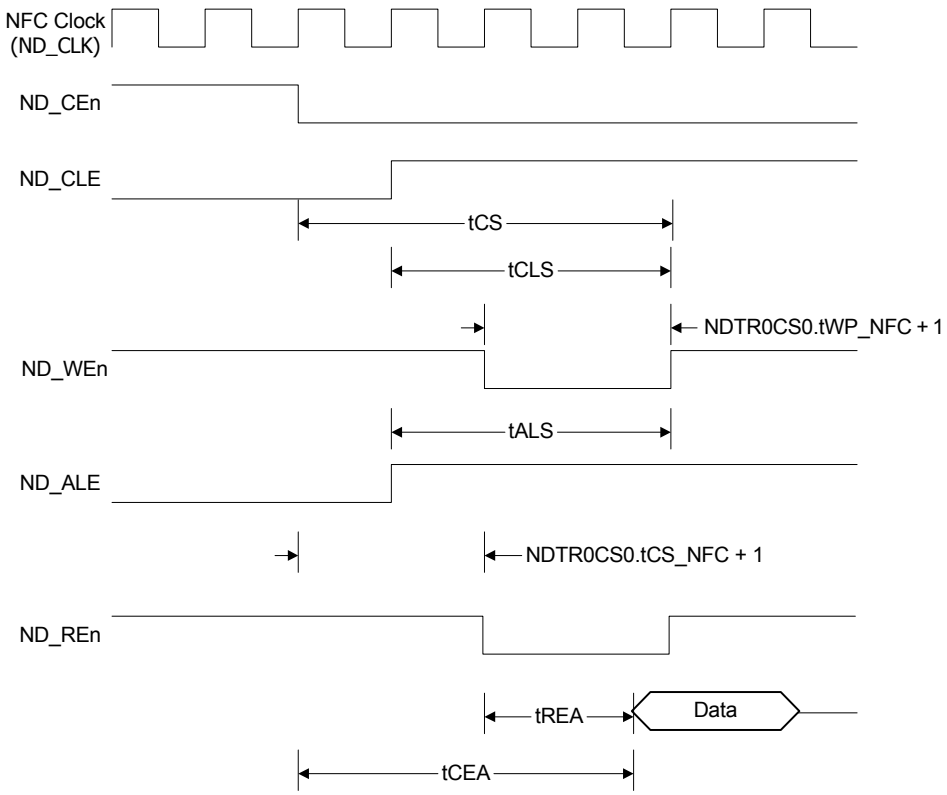
In addition there are two other parameters tCLS and tALS that are the CLE and ALE setup times, respectively, to the rising edge of ND_WEn. The NDTR0CS0.tCS parameter is also used to specify these values, again as the positive difference between tWP and tCLS and/or tALS, whichever is larger.

The last usage of this parameter is to meet the data valid time from CS when NDCR.FORCE_CSX is true. The NAND flash devices have a parameter tCEA that is the data access time from the activating edge of ND_CEn. The read data is valid tCEA, after falling edge of ND_CEn and tREA after the falling edge of ND_REn. The NDTR0CS0.tCS parameter is used to specify tCEA again as the positive difference between tREA (which is related to tRP) and tCEA.

The actual value needed for the NDTR0CS0.tCS parameter is the maximum of whichever values are needed to satisfy tCS, tCLS, tALS, and tCEA (when NDCR.FORCE_CSX is true).

Figure 23 is representative of all of the timing possibilities but not accurate logically as address, command, and data cycles occur at different times and of course it cannot be the case that both ND_REn and ND_WEn are simultaneously asserted.

Figure 23: tCS Timing Example



10.5.2.4 tCH

NAND flash devices have timing parameters that specify the hold time of the control signals (ND_CEn, ND_CLE, ND_ALE) relative to the rising edge of ND_WEn. They are tCH, tCLH and tALH. There is no specific usage of the NDTR0CS0.tCH value in isolation as the actual constraint is the maximum of tCH, tCLH, tALH, and tWH.

The NDTR0CS0.tCH value is set to be the larger of the NAND flash tCH, tCLH, and tALH. The actual hold time implemented by the NFCv2 is the maximum of NDTR0CS0.tCH or NDTR0CS0.tWH.

10.5.2.5 tADL

For program operations, the tADL parameter specifies the time between the rising edge of ND_WEn of the last address cycle and the rising edge of ND_WEn of the first data cycle. There are internal delays between the last address cycle and the first write cycle. In addition, a maximum value of NDTR0CS0.tWH_NFC or NDTR0CS0.tCH_NFC is part of the data cycle. The NDTR0CS0.tADL_NFC field in the NAND Interface Timing Parameter 0 Register (NDTR0CS0) (Table 291 p. 524) must be set to the difference between the NAND device tADL requirement and the Max(NDTR0CS0.tWH_NFC,NDTR0CS0.tCH_NFC) * t(ND_CLK) ns.

10.5.2.6 tAR and tWHR

The tAR parameter is the setup time between the falling edge of ND_ALE and the falling edge of ND_REn for a read cycle. The only time the NFCv2 encounters this case is for a Read ID where this timing parameter must be met between the Read ID address and the first ID data value.

The tWHR parameter is the write hold time between the rising edge of ND_WEn and the falling edge of ND_REn specifically for status reads.

To the NFCv2, a Read ID is similar to a status read and different from a page read, because neither a Read ID nor a status read cycle wait for RBn.

The parameters must be set to one less than the total required time (in NFC clock units).

10.5.2.7 tR

There are multiple ways in which the tR parameter is used. The most obvious is of course to match the NAND flash device tR parameter. If NDTR1CS0.WAIT_MODE is false, a read will wait the time specified in NDTR1CS0.tR_NFC and then proceed to generate read data cycles. In this case, it is important that tR be programmed to meet or exceed the actual NAND flash device tR parameter. If NDTR1CS0.WAIT_MODE is true, a read will wait at least the time specified in NDTR1CS0.tR, and then may continue to wait until NDSR.RDYx actually asserts. In this case, NDTR1CS0.tR_NFC must be programmed to meet or exceed the NAND flash device tWB parameter that is the delay between the rising edge of ND_WEn and the falling edge of ND_RBn.

There is a case that is described in more detail in overlapped program operations where one can use the tR value to simply act as a delay. A “naked read” waits for tR and then perform zero or more data cycles. The key is the zero, because if NDCB0.LEN_OVRD is true and NDCB3.NDLENCNT is set to zero a “zero length naked read” will act as a simple timing delay. This solves a particular problem with overlapped multi-plane, multi-device writes, as described in Section 10.5.6, Program (Write) Usage, on page 137.

Copyright © 2015 Marvell

10.5.3 Entering the First Command

There are two operations to be performed to enter a first command:

- The NFCv2 must be started (that is, NDCR.ND_RUN must be set)
- A three or four word command must be written to the command FIFO at NDCB0.

If the NFCv2 is idle, these operations can be performed in either sequence. If the ND_RUN is set first, the software should wait until NDSR.WRCMDREQ is TRUE. If the NFCv2 is idle, the command buffer will be ready for a command immediately, and the first test should return TRUE. However, it should never be assumed that (in ND_RUN mode) the NFCv2 is ready for a command unless NDSR.WRCMDREQ is TRUE. Once WRCMDREQ is TRUE, it must be cleared in NDSR.

If the NFCv2 is idle, the NDCB FIFO can be first written with the command and then NDCR.ND_RUN set. In this case the software does *not* wait for NDSR.WRCMDREQ before writing the NDCB FIFO. If the command FIFO contains a command and NDCR.ND_RUN is set, the command immediately begins to sequence its execution. For a write type command, NDSR.WRDREQ will set indicating that the NFCv2 is ready for and awaiting data to be written to NDDB. For any other command, the command begins to sequence on the DFI, and for read type commands, NDSR.RDREQ will be set later, after data has been written into NDDB (see Figure 36).

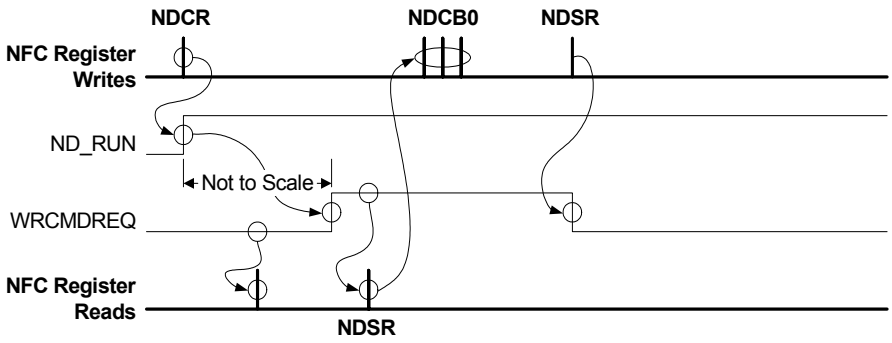
Note

In Figure 24 through Figure 27:

- A thick line signifies an event caused by a processor operation
- A narrow line indicates an internal NFC state
- A circle on a vertical edge indicates an event that causes a state change (where the arrow indicates the cause-effect relation)
- A circle on a horizontal line indicates a sampling operation (where the arrow indicates the reason for the sample).

These diagrams indicate the sequence of operations and events, and are not intended to convey absolute time.

Figure 24: First Command Event Causality Diagram



What happens next depends on the actual command. A reset or erase command will wait for NDSR.CMDD, because there is no data transfer. A read (Read ID, Status) command will wait for NDSR.DTREQ to indicate that data is ready to be read. After reading all data, it should be true that NDSR.CMDD is also set. A program command will wait for NDSR.WRDREQ to indicate that data is ready to be written, and then will wait for NDSR.CMDD.

Copyright © 2015 Marvell

It is not necessary for software to poll the status register to observe status state changes, a software program may be interrupt driven. What follows is representative of a polled software driven usage model presented for the purpose of example and is not intended to be canonical.

10.5.4 Reset Usage

A reset semantic is a single command cycle followed by NDSR.RDYx sensing. For this semantic, the value of the following commands are Don't Care:

- NDCB0.CMD2
- NDCB0.ADDR_CYC
- NDCB0.DBC,
- NDCB0.ST_ROW_EN
- NDCB0.LEN_OVRD
- NDCB0.CMD_XTYPE
- NDCB1
- NDCB2
- or optionally NDCB3

However, it would be preferable to leave all of these fields as zero. For a reset semantic the only fields that matter are:

- NDCB0.CMD1
- NDTR1CS0.tR_NFC
- NDTR1CS0.WAIT_MODE

NDCB0.AUTO_RS should be false.

As is true of any semantic, it is not required that a reset semantic actually issue a reset command. So long as the desired operation is a single command cycle followed by NDSR.RDYx sensing, the semantic can be useful.

An example of a non-reset use of this reset semantic would be:

1. Issue a Naked Command cycle of 0x00 (that is, NDBC0.CMD_TYPE == 0b110).
2. Issue a 5-beat Naked Address cycle (that is, NDBC0.CMD_TYPE == 0b111 and NDCB{1,2} contains Address).
3. Issue a Reset command of 0x30.
4. Issue a Naked Read (that is, NDBC0.CMD_TYPE == 0b000 and CMD_XTYPE == 0b001).

This sequence of four commands has the same effect as a "Monolithic Read". The Naked Command issues the first read command cycle (0x00), the Naked Address issues the 5-cycles of read address, the Reset command issues the Read Commit command cycle (0x30) and waits for NDSR.RDYx. The last command actually reads the data from the NAND and requests that NDDB be emptied. While this is a contrived example, it should illustrate the point that semantics can be strung together in novel ways, as they are needed.

The implementation uses the NDTR1CS0.tR_NFC timing value and NDTR1CS0.WAIT_MODE to wait for NDSR.RDYx.

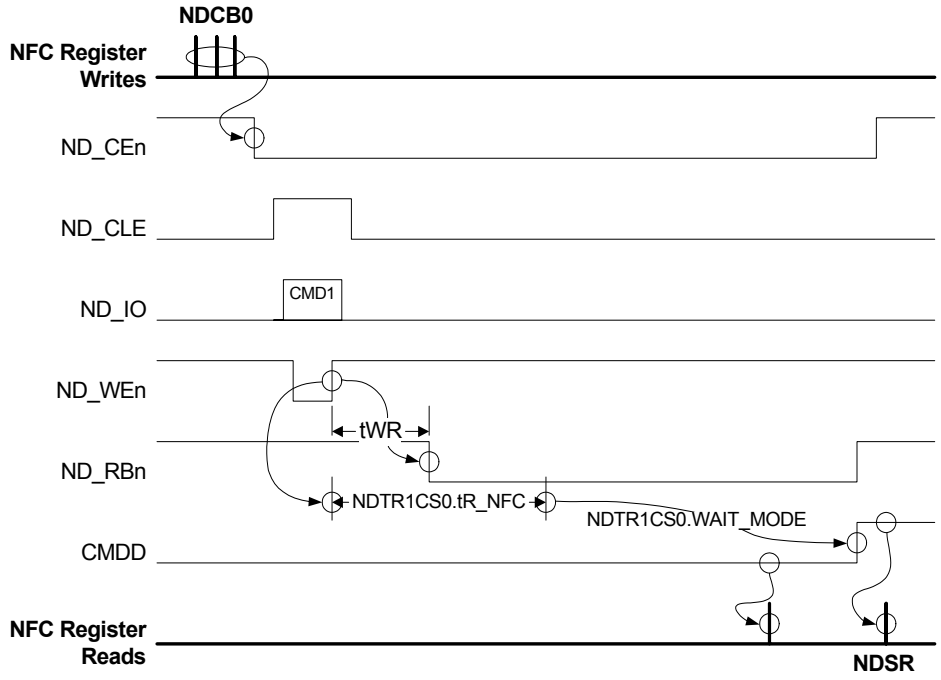
The busy time for a Reset command can vary considerably, from microseconds to milliseconds, depending on the exact circumstances and individual NAND flash devices. The NDTR1CS0.tR_NFC register has a limited dynamic range since it is only 16 bits, and can only measure 420 µs, at the maximum value. To implement the reset in the initial platform case, set the NDTR1CS0.tR_NFC register to a value greater than the value listed in the NAND flash datasheet for tWB (the time from the rising edge of ND_WEn to the falling edge of ND_RBn) and set NDTR1CS0.WAIT_MODE. The software should wait for CMDD to become true. When CMDD becomes true, if NDSR.RDYx has been set, there is a NAND flash present, because it cycled ND_RBn, and if NDSR.RDYx has not been set, a NAND flash is not present in that chip enable position. In this way, the NFCv2 is guaranteed to execute the Reset command to completion, in the NAND present case and is guaranteed to correctly sense the absence of a NAND flash device.

It is possible that the ND_RBn is not available. In this case, the state of ND_RBn is not reliable and cannot be used to determine the end of the Reset command. In this case, if NDTR1CS0.WAIT_MODE is reset, tR does not have enough dynamic range, and if NDTR1CS0.WAIT_MODE is set, the NFCv2 may lock while waiting for a NDSR.RDYx indication that it will never receive. For this reason, a NDTR1CS0.PRESCALE has been added that extends the dynamic range of the NDTR1CS0.tR_NFC value by 16. When NDTR1CS0.PRESCALE is used, the

Copyright © 2015 Marvell
Document Classification: Proprietary Information

effective range of the NDTR1CS0.tR_NFC register is increased to 6.72 ms. If a Reset instruction based discovery is required on platforms where ND_RBnx is connected to something other than a NAND flash device, it will be necessary to set NDTR1CS0.tR_NFC to its maximum value and to also set NDTR1CS0.PRESCALE. This causes the Reset command to always take 6.7 ms, but it also guarantees that there will not be a lockup. In this case, the state of NDSR.RDYx may or may not be a reliable indication of the presence of the NAND flash, as it depends on exactly what the ND_RBn pin is connected to and whether or not it is possible for that pin to assert a positive edge during the Reset command (see Figure 25).

Figure 25: Reset Command Event Causality Diagram



10.5.5 Read ID Usage

A Read ID semantic is a CMD1 command cycle, 0–7 address cycles, a max(NDTR1CS0.tAR,NDTR1CS0.tWHR) timing wait and 0–7 data cycles. There is no RBn wait. Required fields are:

- NDCB0.CMD1
- NDCB0.ADDR_CYC and
- Either NDCR.RD_ID_CNT or NDCB0.LEN_OVRD and NDCB3.NDLENCNT

The following fields are Don't Care but should be set to zero:

- NDCB0.CMD_XTYPE
- NDCB0.ST_ROW_EN
- NDCB0.DBC
- NDCB0.CMD2

NDCB0.AUTO_RS should be false. The number of required ADDR fields in NDCB{1,2} are a function of the ADDR_CYC value.

The most obvious use for a Read ID is to read the NAND ID. In addition, for NAND flash devices that are ONFI compliant, the Read ID can be used to determine this compliance. In the normal ID case, there is a single 0x00 address cycle and either 2, 4, or 5 data cycles. For the ONFI ID case, the address cycle is set to 0x20, and there will be 4 data cycles: O, N, F, and I.

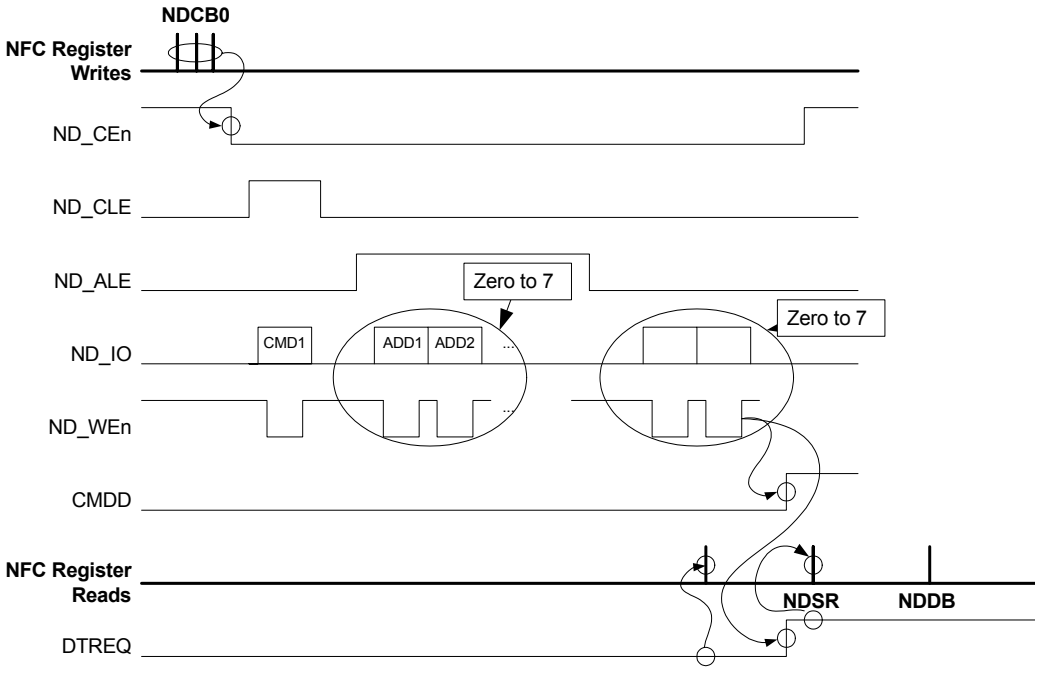
Another possible usage model for the Read ID is for block unlock commands on NAND flash devices that support the lock, lock-tight, and unlock region commands. The unlock command is typically a command cycle followed by three block address cycles to specify the start of the unlocked region, followed by another command cycle with three block address cycles to specify the end of the unlocked region. There should be no data cycles; so to use this semantic as a unlock region command would require that the NDCR.RD_ID_CNT field be set to zero.

There are certain NAND flash devices (or NAND flash device specifications) that have command-address sequences where the single command is a function, and the single address is like an argument for the function. These sequences can be generated as a Read ID with NDCR.RD_ID_CNT set to zero.

Note

- The address cycle count and data cycle counts are now allowed to be zero. If there are zero address cycles specified, no address cycles are generated, and ND_ALE does not assert. If there are zero data cycles specified, no data cycles are generated. Previously the minimum value for either of these fields was defined to be 1, but if these either or both of these fields were set to zero, a minimum of 1 cycle of each type would have been generated.
- For Read ID commands, the software must disable ECC. Clear the <ECC_EN> field in the Data Flash Control Register (NDCR) (Table 290 p. 518) and the <BCH_EN> field in the NAND ECC Control (NDECCCTRL) Register (Table 296 p. 536).

Figure 26: Read ID Command Event Causality Diagram



Copyright © 2015 Marvell

10.5.6 Program (Write) Usage

The program command can be implemented in a number of ways, there are rules for overlapping operations. Now there is support for larger pages and ganging, and, in general, the issues with program operations have become more complex.

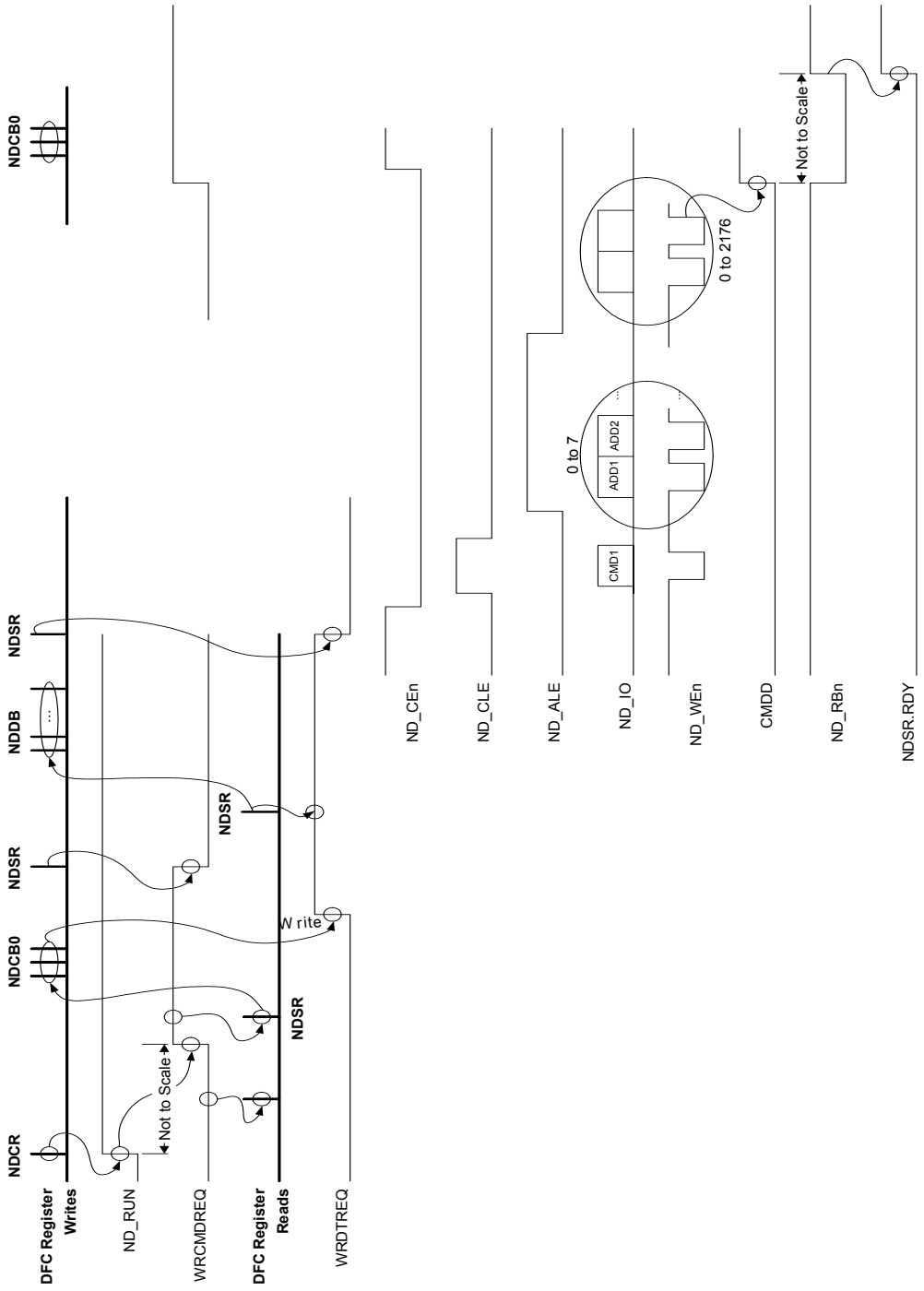
A program is a program command, address, write data, and program commit sequence. A write completes its operation in the NFCv2 once the program commit command is issued (that is, NDSR.CMDD sets). It is possible to overlap another program or erase command with a previous program or erase command if NDCB0.NC is true, NDCB0.RDY_BYP is false, and the second program or erase is to a different CE. A first program or erase command starts by setting NDCR.RUN, waiting for NDSR.WRTCMDREQ, and writing NDCBx with the program command values. Since this is a program, data is required, so next wait for NDSR.WRDREQ. Once the data has been written, the program cycle is sequenced to the NAND flash device. After the command has been sequenced to the NAND flash device, NDSR.CMDDx is set to indicate that the NFCv2 is ready for another command. If NDCB0.NC is true, the NFCv2 sets NDSR.WRCMDREQ and waits for the processor to write in a second command. If the CEs of the 2 operations specify the same NAND flash device and NDCB0.RDY_BYP is false, the second command stalls while waiting for NDSR.RDYx. If the CEs of the 2 operations are different and the second command is a write or erase, the NFCv2 continues to process the command, assuming that the RDY indication of this other chip is not also indicating busy.

After the NAND flash has finished the program cycle, it will assert ND_RBn to indicate ready. NDSR.RDYx sets on the rising edge of ND_RBn.

Figure 27 shows an event causality diagram where a program command is being entered, and optionally if NDCB0.NC is set, a second command is entered for overlap.

Copyright © 2015 Marvell

Figure 27: Write Command Event Causality Diagram



Copyright © 2015 Marvell

Programming Larger Pages

To support larger page sizes, the program command needs to be broken into two commands, using Naked semantics or otherwise known as <CMD_XTYPE> field in the NAND Controller Command Buffer 0 (NDCB0) Register (Table 299 p. 538).

The data transfer portion of the program command is at most 2112 bytes (or 2110 bytes with BCH).

To program a 4 KB page device, the software can do one of the following:

Option 1

1. Issue the initial command or command dispatch: <CMD_XTYPE> = 110.
2. Issue Naked write command: <CMD_XTYPE> = 101.
   and
   Transfer up to 2112 bytes.
3. Issue a second Naked write command and transfer up to 2112 bytes.
4. Issue the final command: <CMD_XTYPE> = b011.

Option 2

1. Issue the initial command or command dispatch with write: <CMD_XTYPE> = 100.
   and transfer the command, address, up to 2112 bytes data.
2. Issue Naked write command <CMD_XTYPE> = 101.
   and
   Transfer up to 2112 bytes.
3. Issue the final command <CMD_XTYPE> = 011.

Option 3

1. Issue the initial command or command dispatch with a write: <CMD_XTYPE> = 100.
   and
   Transfer the command, address, 2112 bytes of data.
2. Issue Naked write with final command: <CMD_XTYPE> = 001.
   and
   Transfer another 2112 bytes, as well as the final confirm command.

The data transfer portion of the program command is at most 2112 (or 2110 with BCH) bytes. To support larger page sizes, the program command must be broken into two commands using “naked” semantics:

- A first program command can issue the program command, address and 2112B of write data.
- A second program command can issue 2112B of write data and the commit command.

The program is composed of two commands, but the total data transfer will support 4-KB page sizes. In the event that 4-KB page NAND flash devices are ganged to create a logical 8-KB page, this may be supported by:

- A first program command that issues the program command, the address and 2112B of write data.
- A second program command can issue 2112B of write data.
- A third program command can issue 2112B of write data.
- A fourth program command can issue 2112B of write data and the commit command.

Naked data writes can be performed to enable any page size. The only caveat is that the data format always has the ECC associated with each data chunk (up to 2 KB).

Copyright © 2015 Marvell

Read Usage

The Read command can be implemented in a number of ways and it has supports for larger pages and ganging, and in general the issues with read operations have become more complex.

A Read semantic is an initial command, address, confirm, or commit command (only for devices that support double byte reads), NAND device pulling RBn Low followed by data transfer. A Read command is considered complete in the NFCv2 only after the read data in the NAND Controller Data Buffer (NDDB) Register (Table 298 p. 537) is read by the CPU.

Reading Larger Pages

To support larger page sizes, the read command needs to be broken down using Extended semantics or Naked semantics.

To read a 4/8 KB page, software can preform the following steps:

1. Issue command dispatch: <CMD_XTYPE> = 110.
2. Issue a naked read command: <CMD_XTYPE> = 101).
   and
   Read 2 KB bytes.
3. Finally issue the consecutive naked read.
   and
   Read the second 2 KB chunk.

Note

- After every 2 KB read from the device the NAND Controller Data Buffer (NDDB) Register needs to be drained.
- To support even larger page sizes, software must perform multiple 2 KB reads.

Naked Command and Naked Address

There are cases where the command and address sequences are not standard or represent new features. In this case, the NFCv2 may not have built-in commands to perform the function(s).

- The ability to issue a naked command means that any NFCv2 operation that depends on CLE being true can be issued.
- The naked address semantic means any sequence of ALE cycles (from 1 to 7) can be issued.
- A naked read cycle can read between zero and 2176B as can a naked write.

In this way, sequences of CLE cycles, ALE cycles, and data cycles can strung together in an almost arbitrary sequence. The only limitation is that the CE for the NAND flash device will cycle between each operation. The primary purpose of these new semantics is to support new features, particularly the command sequences associated with logical block accessed NAND flash device.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

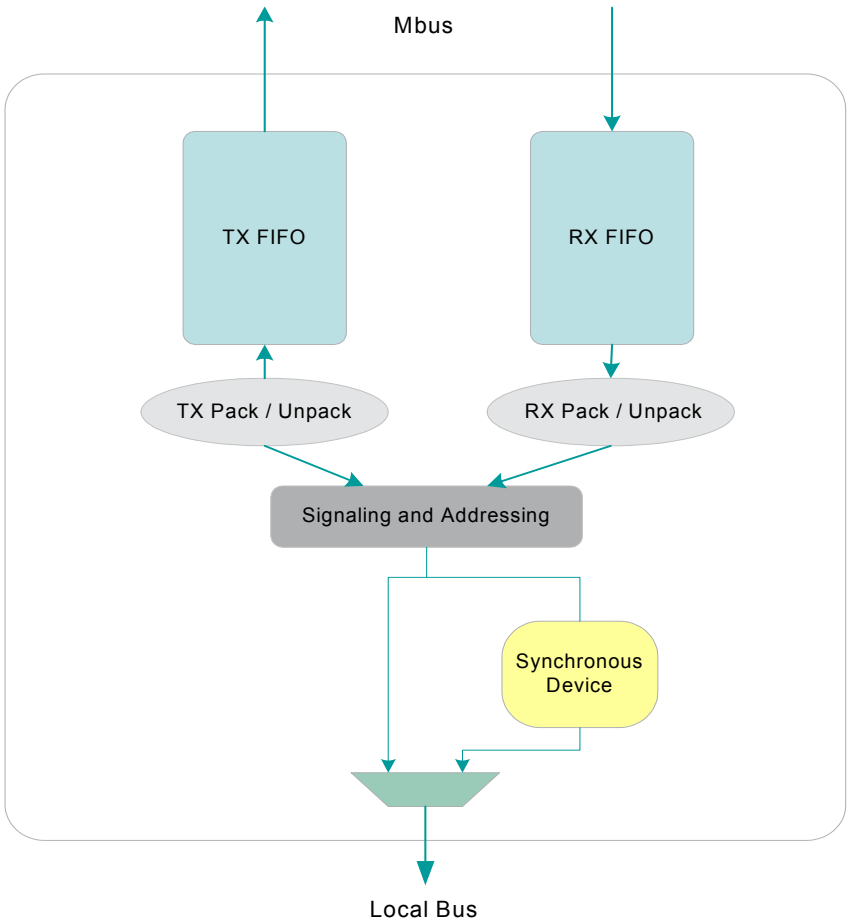
11 Device Bus Controller

This section describes the Device Bus controller. The Device Bus controller is a generic local bus controller that enables the device to interface with off-the-shelf NOR-type flash, SRAMs and custom FPGAs. The section describes the different interface connectivity options, the programmable timing parameters and the interface signaling.

The Device Bus registers are located in Appendix A.5, Device Bus Controller Registers, on page 544.

Figure 28 presents the Device Bus controller block diagram.

Figure 28: Device Bus Controller Block Diagram



11.1 Features

The Device Bus includes the following features:

- Support for different types of standard memory devices, such as Flash and ROM
- 5 chip selects with programmable timing
- Optional external wait-state support
- Up to 128B burst per a single device bus access

11.2 Functional Description

The Device Bus controller supports up to 5 banks of devices. Each bank supports up to 256 MB of address space, and has its own timing parameter register. The bank width can be programmed to 8 and 16 bits. The bank timing parameters can be programmed to support different device types (for example, Flash, ROM, I/O Controllers).

The 5 chip selects are typically separated into 4 individual chip selects (DEV_CSn[3:0]) and 1 chip select for a boot device (DEV_BOOTCSn). The only difference between the boot device bank and any other bank is that, by default, the CPU core boots from the boot device and its default width is sampled at reset.

The Device bus controller supports a multiplexed address/data bus. External latches should be used to latch the address (up to 256 MB of address space). The interface supports up to 128B transfer per a single device access.

Note

When used as 8-bit interface, DEV_AD[15:9] and DEV_WEn[1] can be used for pins multiplexing.

Figure 29 presents the connectivity over the Device bus for devices using a single external latch with address space of up to 256 MB, and OEn/WEn in Shared mode.

Document Classification: Proprietary Information

Figure 29: Device Bus Connectivity for up to 256 MB Address Space (Single External Latch)—OEn/WEn in Shared Mode

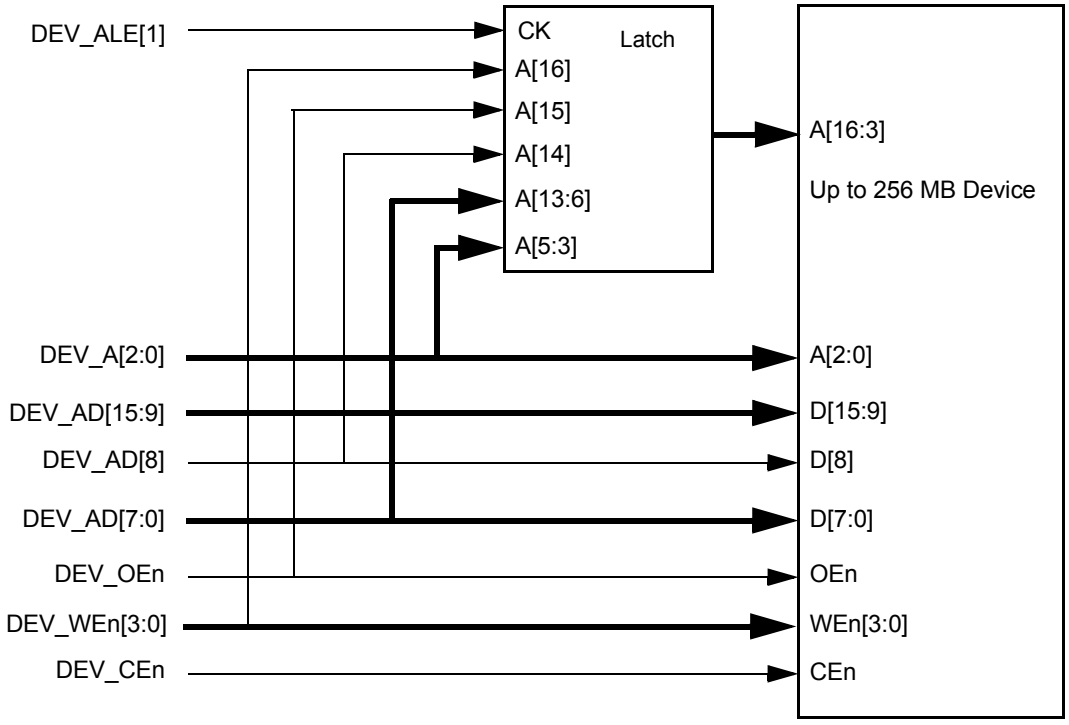
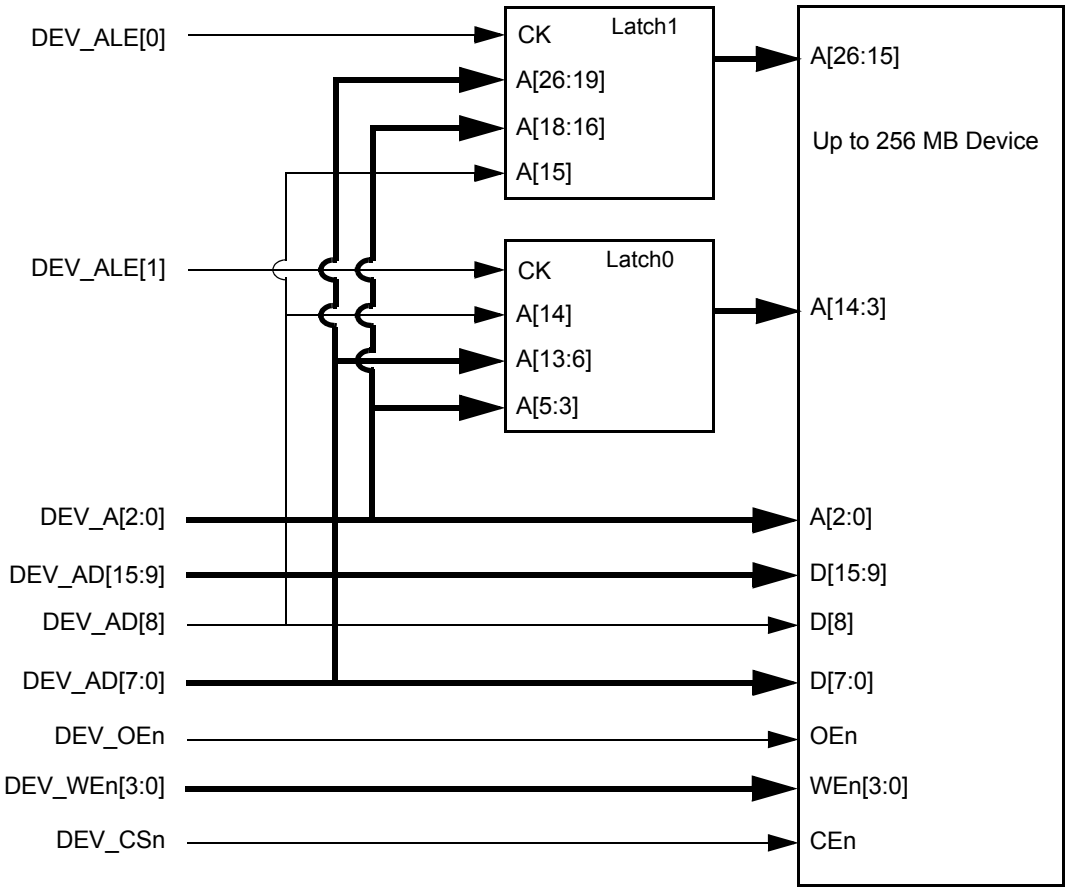


Figure 30 presents the connectivity of devices using 2 external latches over the Device bus with OEn/WEn in Non-Shared mode.

Figure 30: Device Bus Connectivity (2 External Latches)—OEn/WEn in Non-Shared Mode



Copyright © 2015 Marvell

11.3 Address Multiplexing

Figure 31 presents a diagram of the address multiplexing in1 latch mode.

- OEn/WEn shown are in Shared mode, meaning A[15] and A[16] are from OEn and WEn[0].
- There is 1 address phases (ALE[1]) followed by a data phase.

Note In some of the figures that follow, the signal names appear without the DEV_ prefix.

Figure 31: Address Multiplexing with 1 Latch and OEn/WEn in Shared Mode

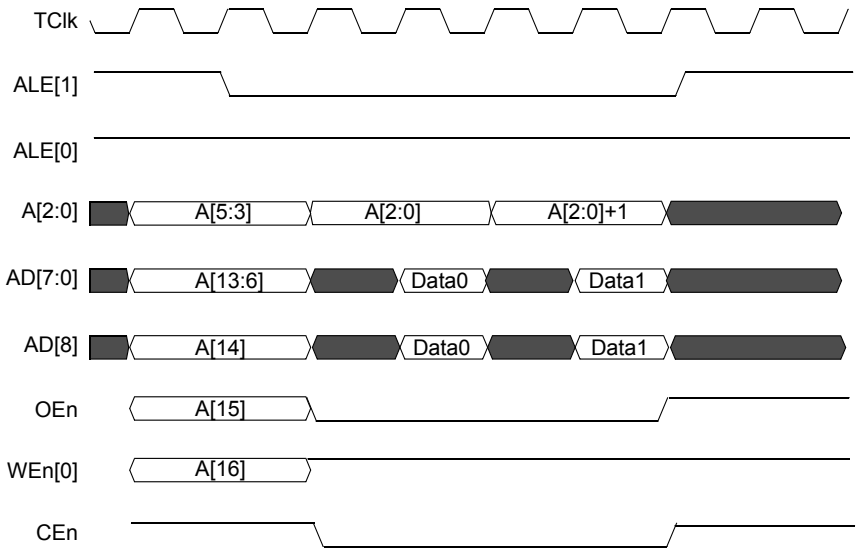
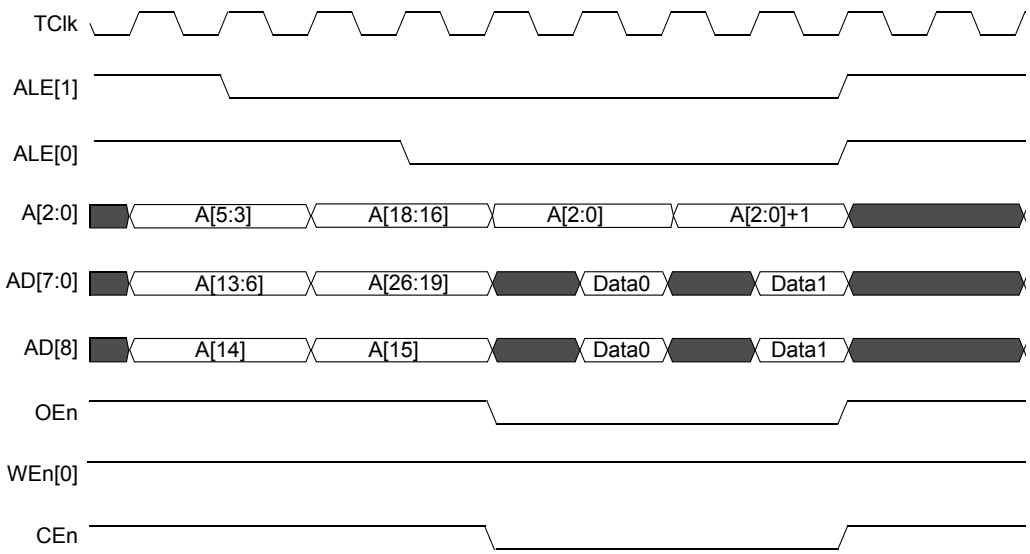


Figure 32 presents a diagram of the address multiplexing in 2 latch mode.

- OEn/WEn must be in Non-Shared mode, so A[15] and A[16] are from A/AD.
- OEn and WEn[0] are at 1 throughout the address phase.
- There are 2 address phases (ALE[1:0]) followed by a data phase.

Figure 32: Address Multiplexing with 2 Latches and OEn/WEn in Non-Shared Mode



Note

- DEV_OEn and DEV_WEn are inactive (high) when the Device bus in idle (that is, from DEV_CSn de-assertion at the end of a transaction until the next transaction DEV_ALE[1] assertion).
- If the <OEWE Shared> field in the Device Bus Interface Control Register (Table 308 p. 547) is cleared to 0, DEV_OEn and DEV_WEn are kept inactive from DEV_CSn de-assertion at the end of a transaction until next transaction's DEV_CSn assertion. This means that for a 2-latch scheme, <OEn/WEn_shared> must be 0x0. To support boot from Device in the desired configuration, <OEn/WEn_shared> is sampled at reset.

Copyright © 2015 Marvell

11.4 Device Bus Timing

11.4.1 Device Bus Address Phase Timing

As described, a transaction address phase consists of two parts—latching of low address bits followed by latching of high address bits.

In an optimal utilization configuration, each of these parts is 2 TCLK cycles long, with DEV_ALE asserted after 1 cycle. This timing results in 1 TCLK cycle setup time and 1 TCLK cycle hold time of address in respect to ALE fall. Since DEV_ALE is loaded with a single latch device, while DEV_AD might be heavily loaded, the actual setup time can become shorter and not sufficient. To solve this issue, the device implements the <Addr2ALE> field in the Device Bus Interface Control Register (Table 308 p. 547) to define the number of cycles in which the address is steady before any ALE assertions.



Note

Since the NOR/NAND hardware arbiter is always enabled by default, the actual setup time is 8 clock cycles longer than configured in the <Add2ALE>.

11.4.2 Device Bus Read Timing Parameters

To allow flexible interfacing to slow and fast devices, the interface can be programmed with different timing parameters for each Device bus chip select.

TurnOff

The <TurnOff> parameter defines the number of TCLK cycles that the device does not drive the AD bus after the completion of a Device read. This prevents contentions on the Device bus after a read cycle from a slow device. The actual number of TCLK cycles from DEV_CSn de-assertion to next DEV_ALE[1] negation is <TurnOff> - <RdHold>.

The minimum setting for this parameter is <RdHold>+2.

Acc2First

The <Acc2First> parameter defines the number of TCLK cycles from the negation of ALE[0] to the cycle that the first read data is sampled by the device. When the maximum configurable value does not meet the requirements, DEV_READYn can be used to further delay the sample (see Section 11.6, READYn Support, on page 154).

The number of TCLK cycles is <Acc2First>-2*<Addr2Ale>-1.

The minimum setting of this parameter is 2 * <Addr2Ale>+2+<RdSetup>+2.

NOTE: When interfacing a synchronous device, set to a value greater than <Acc2Next> + 4.

Acc2Next

The <Acc2Next> parameter defines the number of TCLK cycles between the cycle that samples data N to the cycle that samples data N+1 (in burst accesses). When the maximum configurable value does not meet the requirements, DEV_READYn can be used to further delay the sample (see Section 11.6, READYn Support, on page 154).

The minimum setting for this parameter is 0x2.

<DevCS<n>AssertDelayEn> (n = [3:0])

The <DevCS<n>AssertDelayEn> parameter defines the number of TCLK cycles between address stable point and CS assertion (see the Device Bus Miscellaneous Timing Parameters Register (Table 313 p. 550)). If cleared (default), DEV_CSn assertion will be done on the same TCLK cycle as setting of DEV_A[2:0]. If set, DEV_A[2:0] will be stable 4 TCLK cycles before DEV_CSn assertion.

NOTE: When enabling the <DevCS<n>AssertDelayEn> parameter and running with a small value for ALE2Wr, DEV_WEn asserts before the CS. Therefore, add the restriction:
(ALE2Wr >= 2*(Addr2ALE+1) + 4).

RdSetup

The <RdSetup> parameter defines the number of TCLK cycles between the end of the address phase and DEV_OEn assertion. If cleared to 0 (default), DEV_CSn assertion, DEV_OEn assertion, and DEV_A[2:0] setting will be occur on the same cycle.

To assure compliance with other timing configurations:

1. Set <RdSetup> to a value smaller than <Acc2First>-2*<ALE2Addr>-2.

2. If <CsSetup> is enabled, set <RdSetup> to a value which is greater or equal to 4.

NOTE: RdSetup < ALE2Wr.

RdHold

The <RdHold> parameter defines the number of TCLK cycles between the last data sample to the de-assertion of DEV_CSn. If cleared to 0 (default), DEV_OEn and DEV_CSn are de-asserted at the same cycle (the cycle of the last data sample).

The <RdHold> parameter has no affect on DEV_OEn de-assertion. DEV_OEn is always de-asserted the next cycle after last data sampled.

Also, the <RdHold> parameter has no affect on <TurnOff> parameter. Set <RdHold> to a value smaller than <TurnOff>.

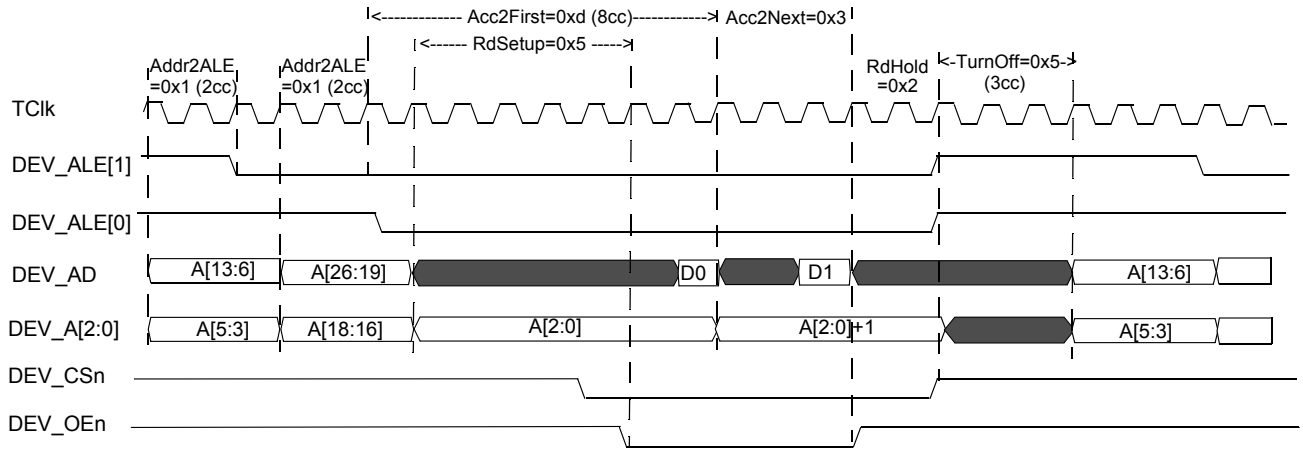
RdHoldADEn

Setting the <RdHoldADEn> field in the Device Bus Miscellaneous Timing Parameters Register (Table 313 p. 550) forces DEV_A[2:0] to remain stable until the de-assertion of DEV_CSn.

Document Classification: Proprietary Information

Figure 33 presents a device read timing parameters example.

Figure 33: Read Parameters Example



11.4.3 Device Bus Write Timing Parameters

To allow flexible interfacing to slow and fast devices, the interface can be programmed with different timing parameters.

ALE2Wr

The <ALE2Wr> field in the DEV_CSn[n] Write Parameters Register (n=0–3) (Table 307 p. 547) defines a guaranteed gap between address phase and DEV_WEn assertion. The number of cycles from DEV_ALE[0] negation to DEV_WEn assertion will be at least<ALE2Wr>-2*<Addr2ALE>-3.

The minimum setting of this parameter is 2*(<Addr2ALE>+1)+2.

NOTE: RdSetup < ALE2Wr.

WrLow

The <WrLow> field in the DEV_CSn[n] Write Parameters Register (n=0–3) (Table 307 p. 547) defines the number of TCLK cycles that DEV_WEn is active (low). Extend this parameter by extending the READYn pin (see Section 11.6, READYn Support, on page 154). DEV_A[2:0] and Data are kept valid as long as DEV_WEn is active. This parameter defines the setup time of address and data to DEV_WEn rise. The minimum setting of this parameter is 0x1.

WrHigh

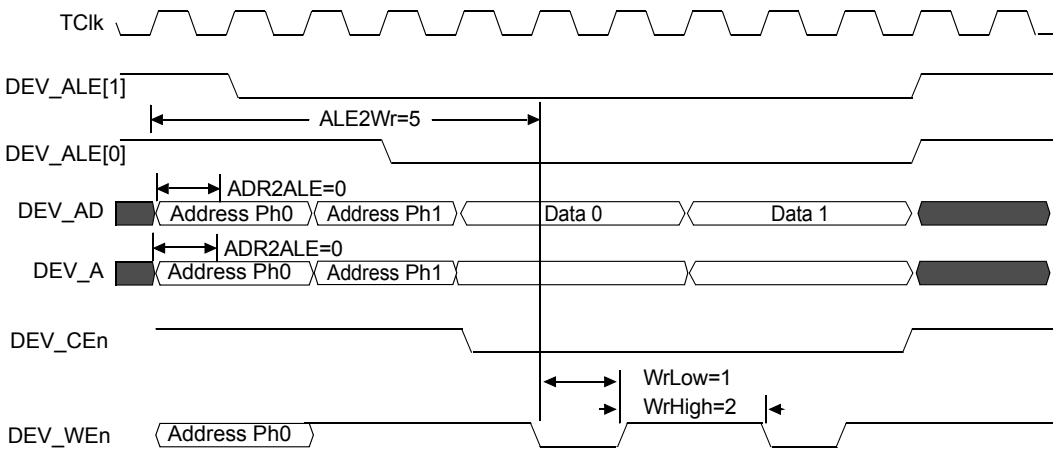
The <WrHigh> field in the DEV_CSn[n] Write Parameters Register (n=0–3) (Table 307 p. 547) defines the number of TCLK cycles that DEV_WEn is kept inactive (high) between data beats of a burst write. DEV_A[2:0] and Data are kept valid (do not toggle) for <WrHigh>-1 periods. This parameter defines the hold time of address and data after DEV_WEn rise. The minimum setting of this parameter is 0x1.

If setting <WrHigh> to 1, DEV_A[2:0] and write data are toggled on the same cycle DEV_WEn toggles from High to Low.

Copyright © 2015 Marvell

Figure 34 presents a device write timing parameters example.

Figure 34: Write Parameters Example



The device bus is typically used for interfacing asynchronous devices that do not require clock input. However, it can also be used to interface synchronous devices. The 88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828 can drive this clock on the DEV_CLKOUT pin.

DEV_CLKOUT reflects the internal core clock (TCLK). DEV_CLKOUT = TCLK/4.
DEV_CLKOUT can be configured to drive a divided TCLK via the <DEV_CLKOUT Divide Value> field in the Device Bus Synchronous Control Register (Table 310 p. 549). This can be useful for interfacing synchronous devices that can not run as fast as TCLK frequency.

Note

Device bus signals are driven and sampled on the rising edge of the core clock (TCLK), even when the chip is configured to drive a divided TCLK on DEV_CLKOUT pin.

On odd divider values, the duty cycle of the driven clock will not be 50%.

Interacting with a Synchronous Device

When DEV_CLKOUT is used for interacting with a synchronous device, the timing parameters configurations must correlate to the required clock frequency. In particular, the following restrictions must hold:

- <Acc2First> must be greater than + 4.
- In case of a burst read, DEV_READYn can be used to delay the sampling point of the first data (that is, it can extend <Acc2First>) but it cannot be used to delay the sampling point of the following phases (that is, it cannot extend <Acc2Next>).
- <WrLow> + <WrHigh> must equal <Tclk Divide Value>.
- <ALE2Wr> must be greater than <Tclk Divide Value>.

Copyright © 2015 Marvell

11.4.4 Device Bus Transaction Timing Summary

The following section summarizes the Device Bus configuration options presented in Section 11.4.1, Device Bus Address Phase Timing, on page 147 through Section 11.4.3, Device Bus Write Timing Parameters, on page 149. It provides additional clarification due to inherited dependencies between the defined timing configurations.

The section contains diagrams and tables that provide a method for defining the required minimal durations of non-connected gaps based on system requirements and then applies the configurations to meet these requirements.

11.4.4.1 Read Transaction Timing

Read Timing Gaps

Figure 35 presents a diagram of the Read timing for the NOR interface.

Figure 35: NOR Interface Read Timing Diagram

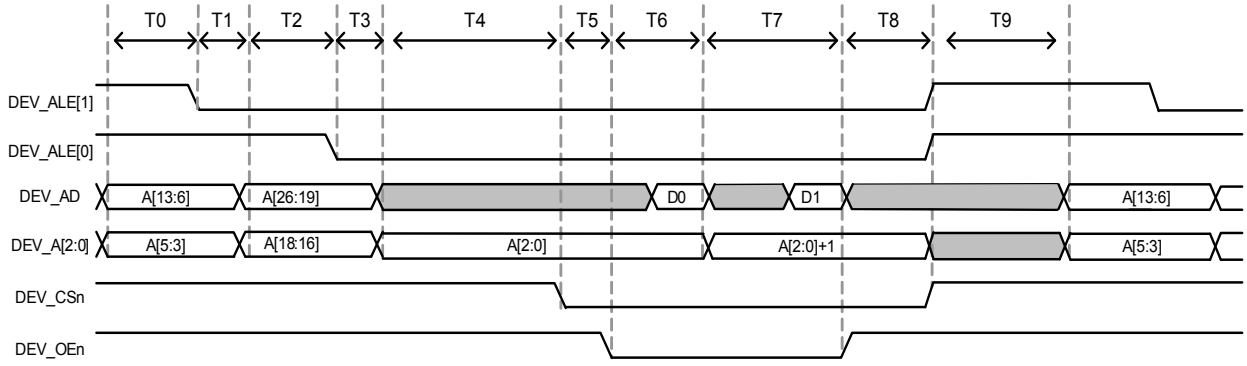


Table 23 provides a description of the timing segments and the possible range of values that can be configured to achieve a guaranteed minimal gap.

Table 23: Read Timing Parameters

Label	Description	Minimum
T0	Number of TCLK cycles from first address phase stable cycle to DEV_ALE[1] negation	1,2,3
T1	Number of TCLK cycles from DEV_ALE[1] negation in which the address remains stable	1
T2	Number of TCLK cycles from the second address phase stable cycle to DEV_ALE[0] negation	1,2,3
T3	Number of TCLK cycles from DEV_ALE[0] negation in which the address remains stable	1
T4	Number of TCLK cycles from the end of the address phase to assertion of DEV_CSn	0,4
T5	Number of TCLK cycles from DEV_CSn assertion to DEV_OEn assertion	0,1..30
T6	Number of TCLK cycles from DEV_OEn assertion to the cycle in which the first data is sampled	1,2..30
T7	Number of TCLK cycles from DEV_A[2:0] toggle to the cycle in which the corresponding data is sampled	1,2..63
T8	Number of TCLK cycles from DEV_OEn de-assertion to DEV_CSn de-assertion	0,1..31
T9	Number of TCLK cycles from DEV_ALE[1:0] release until the first cycle of the address phase of the following transaction	1,2..30

Read Timing Configurations

Given a set of values for T0-T9 with in the configuration as defined in Table 23, the following expressions provide the required configuration:

- <Addr2ALE> = T0–1 = T2–1
- <CsDelay> = If (T4==0), <CsDelay> = 0x0, otherwise <CsDelay> = 0x1
- <RdSetup> = T4 + T5
- <Acc2First> = T0 + 1 + T2 + 1 + T4 + T5 + T6
  When the arbiter is used: 2 + T4 + T5 + T6 - Ready_delay
- <Acc2Next> = T7
- <RdHold> = T8
- <TurnOff> = T9 + T8

Ready Support

An inactive Ready signal delays Acc2First (gap T6) or Acc2Next (gap T7).

11.4.4.2 Write Transaction Timing

Write Timing Gaps

Figure 36 presents a diagram of the Write timing for the NOR interface.

Figure 36: NOR Interface Write Timing Diagram

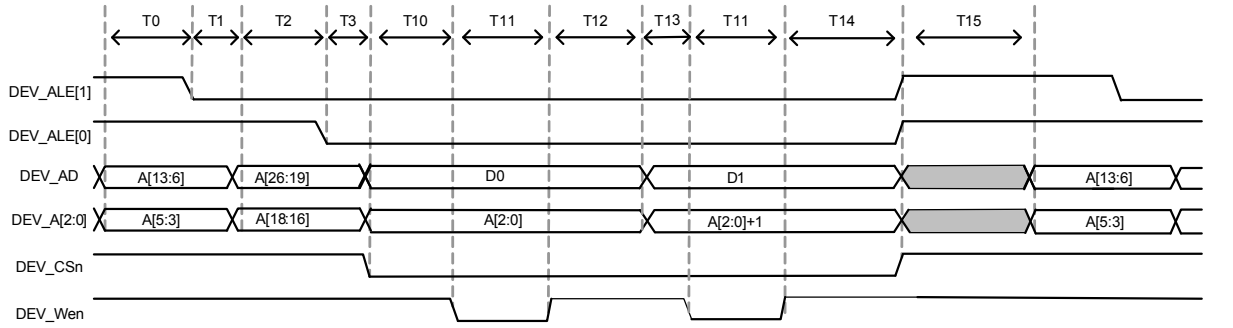


Table 24 provides description of the timing segments and the possible range of values that can be configured to achieve a guaranteed minimal gap.

Table 24: Write Timing Parameters

Label	Description	Minimum
T0	Number of TCLK cycles from the first address phase stable cycle to DEV_ALE[1] negation	1,2,3
T1	Number of TCLK cycles from DEV_ALE[1] negation in which the address remains stable	1
T2	Number of TCLK cycles from the second address phase stable cycle to DEV_ALE[0] negation	1,2,3
T3	Number of TCLK cycles from DEV_ALE[0] negation in which the address remains stable and DEV_CSn is not asserted	1
T10	Number of TCLK cycles from the point that DEV_CSn is asserted and the first address is stable until DEV_WEn assertion	1,2,3..55
T11	Number of TCLK cycles in which DEV_WEn is kept low	1,2,3..63

Copyright © 2015 Marvell

Table 24: Write Timing Parameters (Continued)

Label	Description	Minimum
T12	Number of TCLK cycles from DEV_WEn rise in which the address and data remains stable	0
T13	Number of TCLK cycles from the address and data setting until negation of DEV_WEn	1
T14	Number of TCLK cycles from DEV_WEn last rise until DEV_ALE[1:0] release	1,2,3..63
T15	Number of TCLK cycles from DEV_ALE[1:0] release of a write transaction until the first cycle of the address phase of the following transaction	5

Write Timing Configurations

Given a set of values for T0–T2,T10–T15 with in the configuration as defined in Table 24, the following expressions provide the required configuration:

- <Addr2ALE> = T0–1 = T2–1
- <ALE2Wr> = T0 + 1 + T2 + 1 + T10
- <WrLo> = T11
- <WrHi> = T12+1 = T14 + 1

Ready Support

An inactive Ready signal delays WrLo (gap T11).

Copyright © 2015 Marvell

11.5 Data Pack/Unpack and Burst Support

The Device Bus interface is an 8/16-wide interface and supports up to a 128B burst. The Device Bus controller contains a single 128B buffer for RX and 128B buffer for TX. When a 8b or 16b device is used, Multiple 8B bursts are performed towards the device until the data transfer is completed.

The burst address on the external interface is supported by a dedicated 3-bit A[2:0] signals.

A[2:0] must be connected directly to the device address bus (as opposed to the latched address on the multiplexed AD bus). The Device bus controller supports packing/unpacking of data between the external 8/16-bit device to the internal 64-bit data path.

As previously described, the Device controller needs to pack read data from the 8/16-bit wide Device to the internal 64-bit data path. The Device controller drives the read data to the initiator only when the transaction on the Device bus completes and all data resides on its internal buffer.

11.6 READYn Support

READYn input is used to extend the programmable device timing parameters. This is useful for two cases:

- Interfacing a very slow device that has access time greater than the maximum programmable values
- Interfacing a device with a non deterministic access time (access time depends on other system events and activity)

READYn can extend these timing parameters <Acc2First>, <Acc2Next>, and <WrLow> in the following registers:

- DEV_BOOTCSn Read Parameters Register and DEV_BOOTCSn Write Parameters Register
- DEV_CSn[n] Read Parameters Register (n=0–3) and DEV_CSn[n] Write Parameters Register (n=0–3)

During a read access, the Device controller first counts TCLK cycles based on <Acc2First> programmable parameters. If at the time <Acc2First> is expired, READYn input is not asserted, it keeps waiting until READYn is sampled asserted, and only then samples first read data. Similarly, if at the time <Acc2Next> is expired, READYn is not asserted, it keeps waiting until READYn is sampled asserted, and only then samples next read data. On a write access, if at the time WrLow is expired, READYn input is not asserted, it keeps driving write data until READYn is sampled asserted. Figure 37, Figure 38, and Figure 39 show examples of the READYn operation.

Note

When interfacing a device with a non-deterministic access time, set the timing parameters to their minimum values, and the actual access time is controlled via the READYn pin.

When working in synchronous mode, READY can be used to extend <Acc2First> only (see Section, Interacting with a Synchronous Device, on page 150).

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Figure 37: READYn Extending Acc2First Example

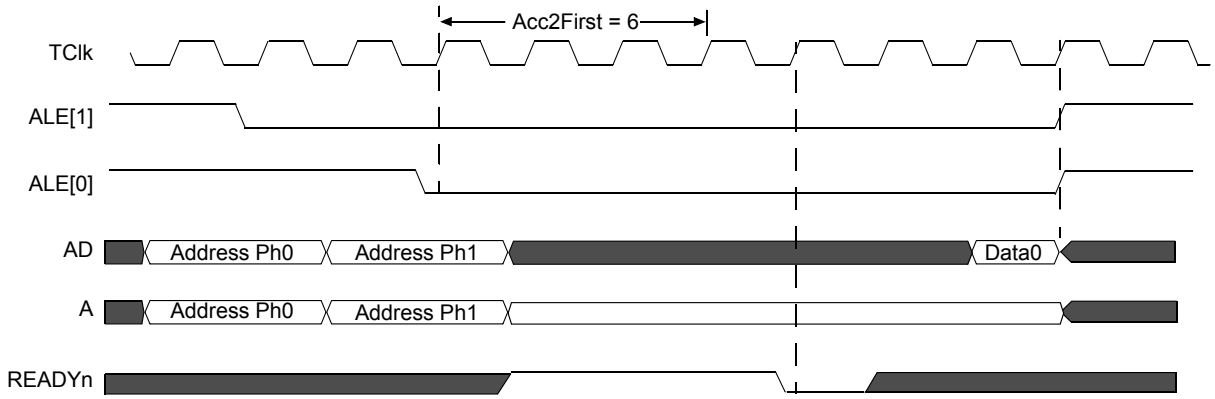


Figure 38: READYn Extending Acc2Next Example

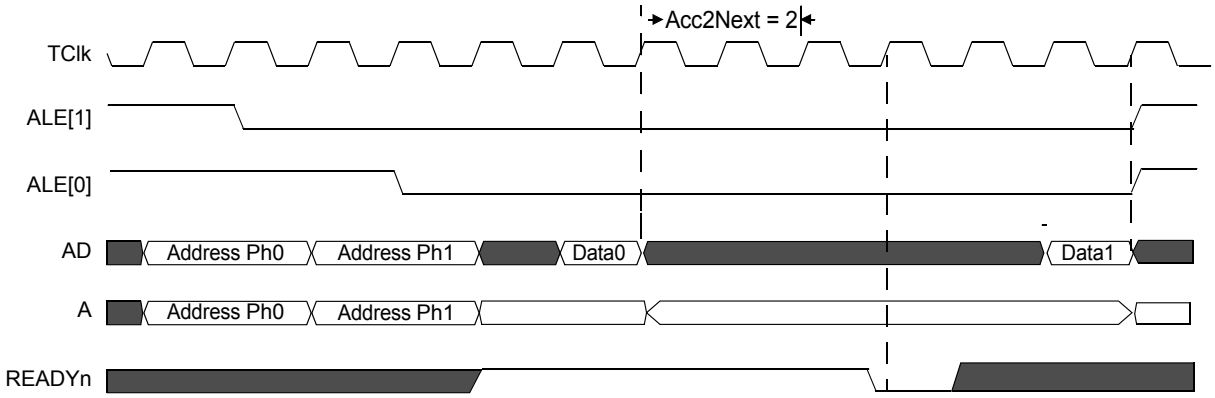
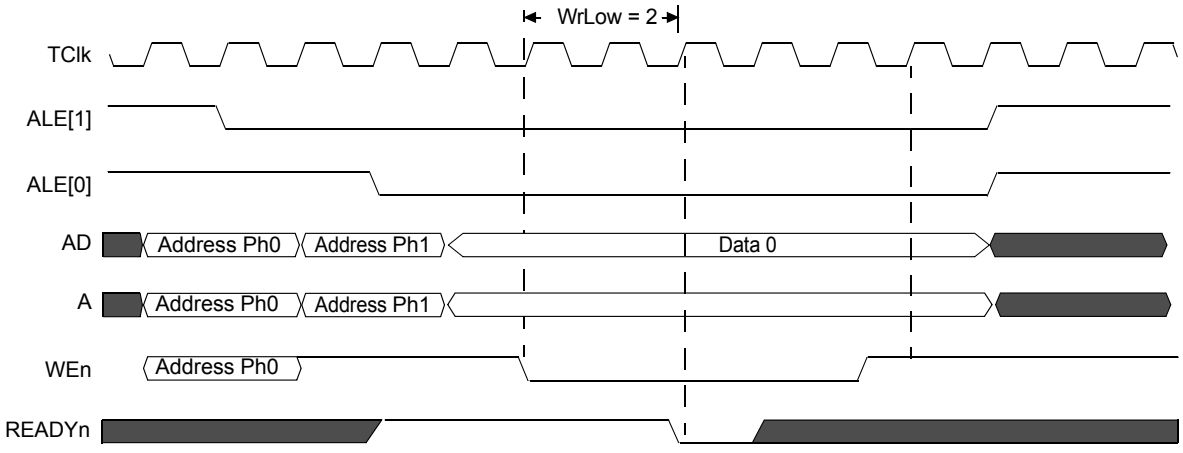


Figure 39: READYn Extending WrLow Example



Copyright © 2015 Marvell

By default, the Device bus controller samples read data after DEV_READYn assertion on a read access.

Each Device bus chip select can be configured to support DEV_READYn signal or not in the CSx Ready ignore fields in the Device Bus Synchronous Control Register (Table 310 p. 548). If configured not to support DEV_READYn, access time is controlled by the timing parameters, If configured to support DEV_READYn, read and write timing is controlled by DEV_READYn assertion as described above.

If interfacing with multiple devices on the Device bus which require DEV_READYn support, external logic is required to qualify the DEV_READYn input with the appropriate device chip selects.

To prevent system hang due to a lack of READYn assertion, the device implements a <Timeout> field in the Device Bus Interface Control Register (Table 308 p. 548) programmable timer. This field allows for a termination of a device access even without READYn assertion.

If during a device access the time-out timer expires, the Device controller completes the transaction as if READYn was asserted, generates an interrupt, and latches the accessed address in <Addr> field in the Device Bus Error Address Register (Table 309 p. 548). This might result in bad data read/write from/to the device. Setting the timer to 0x0 disables the timer, and the Device controller waits for READYn forever.

Note

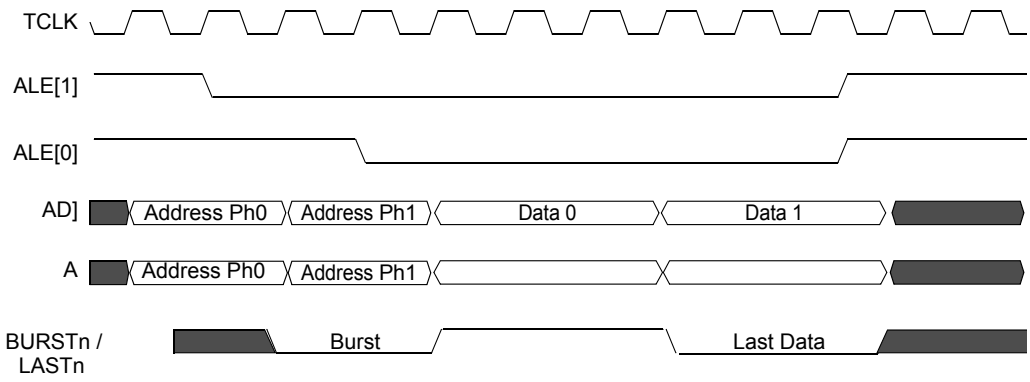
The timer must be programmed to a number that must never be exceeded in normal operation.

11.7 Additional Device Interface Signaling

To make it easy to glue external logic on the Device bus, the device supports a burst and last indication. BURSTn/LASTn is driven low on the address phase to indicate a burst access and is driven low on the last data phase to indicate the last data transfer. It needs to be latched via ALE[0].

Figure 40 presents an example of these additional signals.

Figure 40: BURSTn/DEV_LASTn Example



Copyright © 2015 Marvell



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU System on Chip

Functional Specifications – Unrestricted

Part 4: Ethernet Networking Subsystem

- Section 12, Ethernet Networking Controller
- Section 13, 1G/2.5G Ethernet MAC
- Section 14, Precise Time Protocol (PTP)
- Section 15, Hardware Buffer Management Controller

Doc. No. MV-S109094-U0, Rev. A
November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell

12 Ethernet Networking Controller

This section describes the Ethernet Networking Controller.
Figure 41 presents a block diagram of the Ethernet Networking controller.

Figure 41: Ethernet Networking Controller Block Diagram

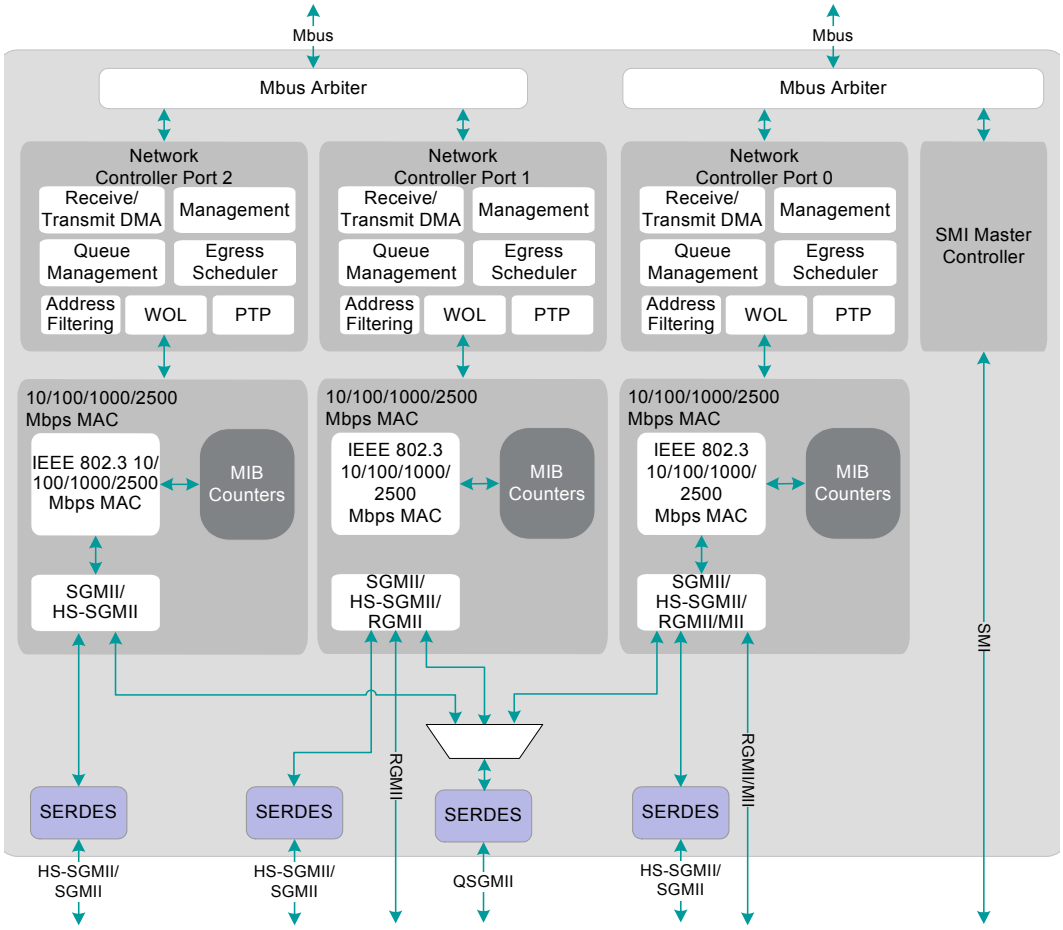
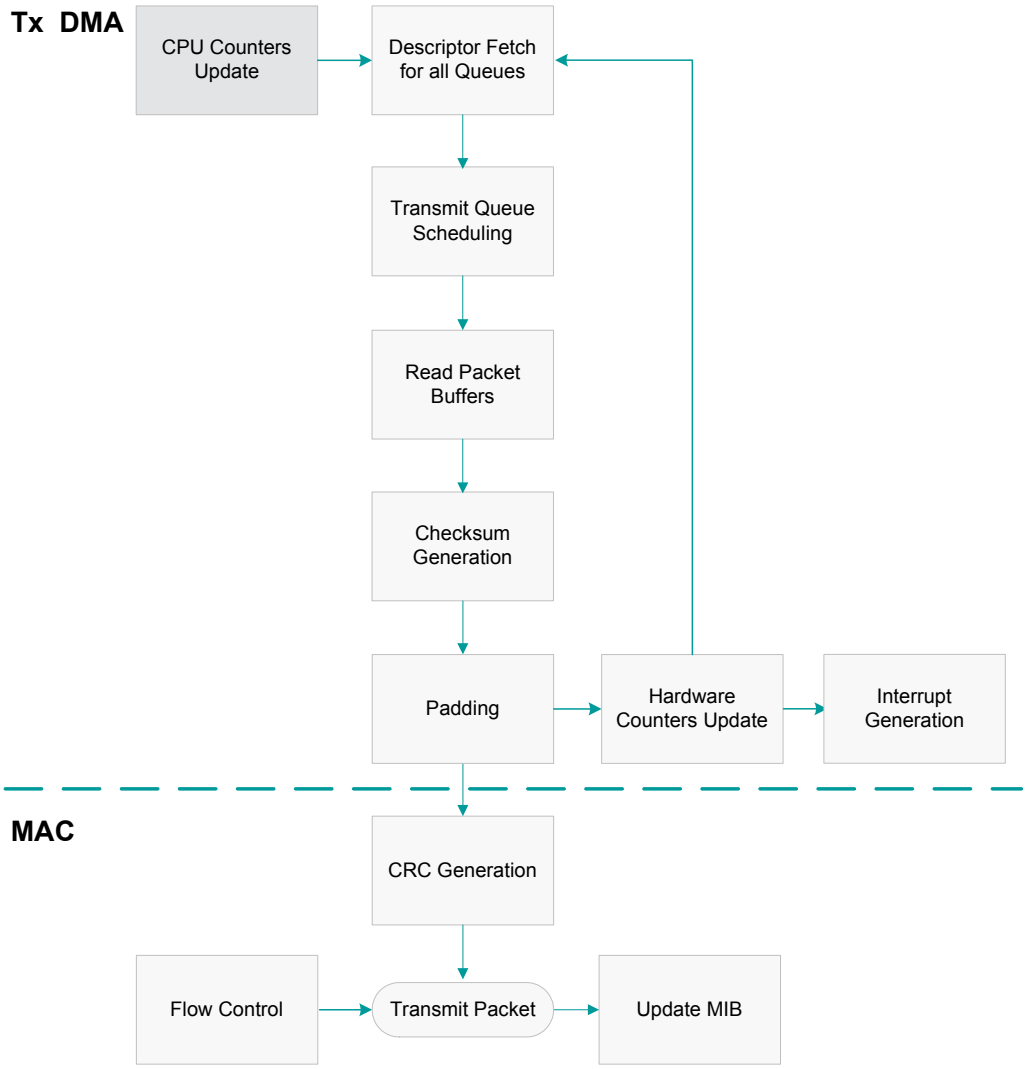


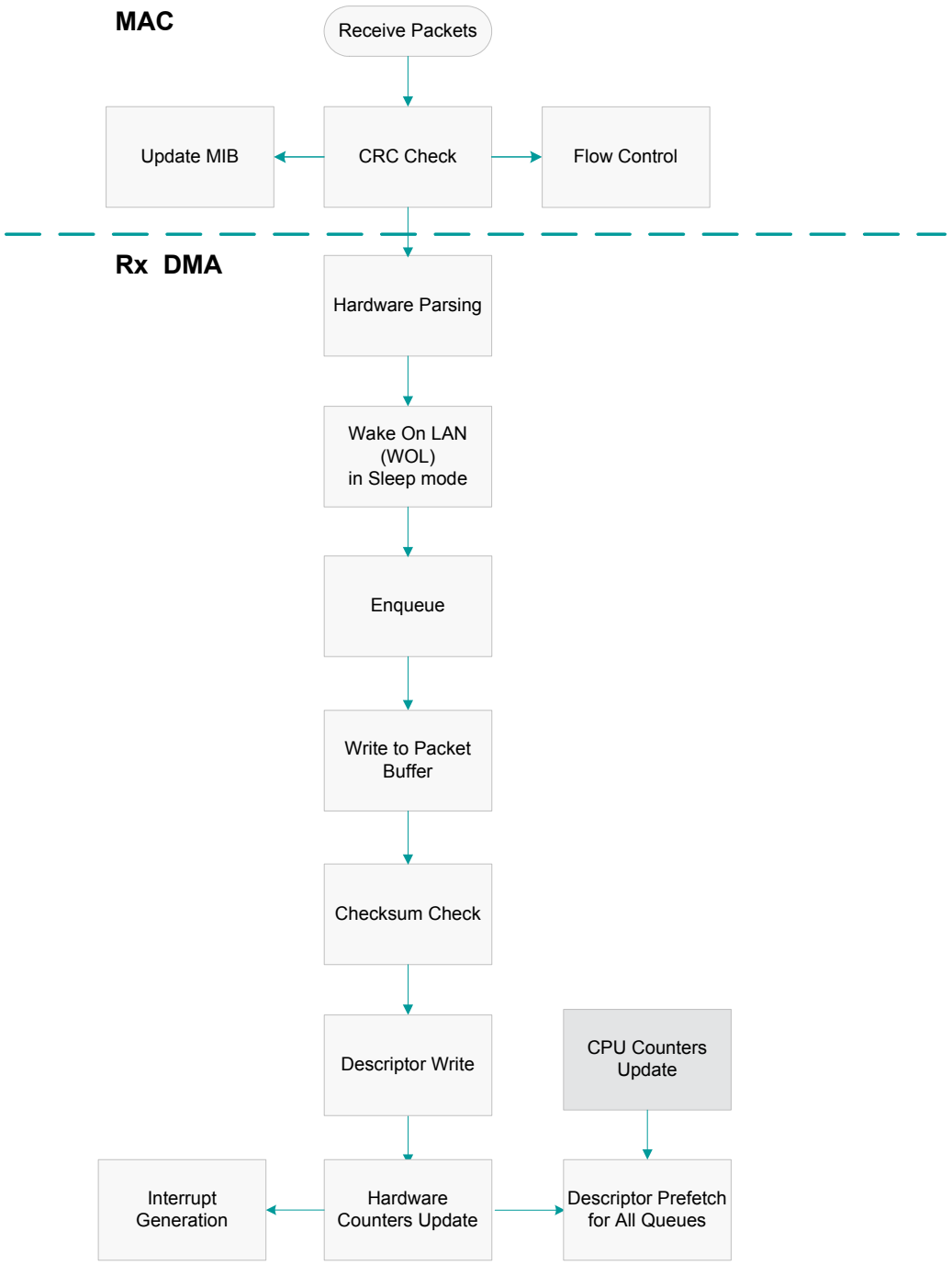
Figure 42 and Figure 43 show the Ethernet Networking controller packet transmit and receive walkthrough, respectively.

Figure 42: Ethernet Networking Controller Transmit Packet Walkthrough



Copyright © 2015 Marvell

Figure 43: Ethernet Networking Controller Receive Packet Walkthrough



The Ethernet Networking cluster supports up to 3 GbE ports and SGMII / QSGMII / RGMII / MII interfaces.

The controller provides the following features:

- The hardware buffer management supports buffer allocation and release without software intervention.
  For ingress packets: The networking controller gets the buffer(s) directly from the Buffer Management (BM) controller.
  For egress packets: The networking controller releases the buffer(s) directly to the BM controller.
- Simple hardware-to-software interface for Rx and Tx queuing management.
  This reduces software overhead and improves performance.
- Receive and transmit queuing is based on a buffer-descriptor list. Descriptors and data transfers are performed by the port-dedicated DMA.
- IPv6 support: TCP/UDP checksum check (RX) and generation (TX), IPv6 based QoS.
- The networking controller leaves configured space in the header of the buffer for software use (for example, to add the VLAN header) by writing the packet to the buffer with an offset.
- Interrupt coalescing for received packets (time and packet based).

12.1 Networking Controller Modes Overview

During initialization, the networking controller mode is configured for the Receive (Rx) and Transmit (Tx) paths by the <AccelerationMode> field in the Port Acceleration Mode (PACC) Register (Table 539 p. 1149).

Queues Management mode
- Queues management by counters

Buffer Management mode options
- Legacy buffer management in which buffers allocation and release is accomplished by the software only.
- Hardware buffer management in which buffers allocation and release is accomplished by the BM controller or the software.

The Descriptor Management by Counter options are described in Section 12.3, Queues Management – Counters Mode, on page 169.

Table 25 lists the coding for the <AccelerationMode> field.

Table 25: Coding for <AccelerationMode>

<Acceleration Mode> Field Value	Queue Management Method	Parsing Mode	Buffer Management Mode
1	Counters	Hardware parsing	Software only
2	Enhanced (by counters)	Hardware parsing	Hardware or software

Document Classification: Proprietary Information

<AccelerationMode> Value	1	2
<AccelerationMode> Feature	Queue: Counter Parsing: Hardware BM: Software	Queue: Counter Parsing: Hardware BM: Hardware/Software
Queues handling method	By counters	By counters
RX DMA leaves configured space in the header of the buffer for software use	Yes	Yes
Buffers management method	Software only	Hardware or software
L2 CRC • Generation (Tx) • Checking (Rx)	Yes	Yes
L3/L4 checksum checking (Rx)	IPv4 TCP/UDP over IPv4	IPv4 TCP/UDP over IPv4
L3/L4 checksum generation (Tx)	IPv4 TCP/UDP over IPv4	IPv4 TCP/UDP over IPv4
Wake On LAN	Yes	Yes
PTP	Yes	Yes
Interrupt coalescing for Rx queues	Time based and descriptors based	Time based and descriptors based
Interrupt coalescing for Tx queues	Descriptors based	Descriptors based
Descriptors format	32B Rx descriptors 16B Tx descriptors	32B Rx descriptors 16B Tx descriptors
Rx snoop support	Yes	Yes
L2 deposit support	Yes	Yes
Port Flow Control	Yes	Yes
Marvell Header/DSA tag	Yes	Yes
Transmit arbitration (strict priority, WRR)	Yes	Yes
Tx Bandwidth limitation resolution	10 Kbps	10 Kbps
Configurable Inter packet gap	Yes	Yes

Copyright © 2015 Marvell

12.2 Functional Description

The networking controller provides Ethernet ports functionality, with the port(s) capable of running at either 10 or 100 Mbps (half or full duplex) and 1000 Mbps (full duplex only). In addition, 2500 Mbps is supported. The networking controller manages packet data transfer between memory and PHY. The data is stored in memory buffers. A packet may be stored in single buffer or multiple buffers (and multiple descriptors, as each descriptor is pointing to a single buffer) if necessary.

In all enhanced modes, IPv4 checksum, TCP over IPv4/v6 checksum, and UDP over IPv4/v6 checksum are always checked on received packets, and may be generated for transmit.

This capability increases performance significantly by off-loading these operations from the software. Checksum is checked for Jumbo frames on received packets. Checksum generation for transmitted Jumbo frames is not supported.

Each networking controller includes eight dedicated receive DMA queues and eight dedicated transmit DMA queues, plus two dedicated DMA engines (one for receive and one for transmit) that operate concurrently.

The networking controller provides programmable zero padding of short transmitted frames (frames that are less than 64B).

Detailed status is given for each receive frame in the packet descriptors, while statistics are accumulated for received and transmitted traffic in the MIB counters, on a per port basis.

The Ethernet controller handles all functionality associated with moving packet data between local memory and the networking interface.

Received frames smaller than the programmable minimum frame size are automatically discarded. Reception and transmission of long frames, up to 9700B, are supported on Port 0. The frame type, encapsulation method, errors, and checksums are reported in the buffer descriptor. Automatic IP header 32-bit alignment is done in memory by adding 2B at the beginning of each frame. The TCP and UDP checksum calculations are put into the receive descriptor (and are compared with the frame checksum for non-IP fragmented frames), even for frames over 9 KB.

Byte based bandwidth distribution among transmit queues by a weighted-round-robin arbitration mechanism is programmable, to provide QoS among the queues. This includes programming of hybrid strict and round-robin priorities among the transmit queues. The maximum byte based bandwidth allocation per transmit queue is also programmable. An Egress Jitter Pacing (EJP) arbitration mechanism is supported, to comply with IEEE 802.1Qav pre-draft, for AVB (Audio Video Bridging).

Zero Padding of Short Frames

The Ethernet networking controller offers a per frame padding enable bit in the transmit descriptor. This causes the networking controller to enlarge frames shorter than 64B by appending zero bytes. When this feature is used, only frames equal or larger than 64B are transmitted as is. Frames smaller than 64B are zero padded and transmitted as 64B packets.



Note

When using the Marvell® Header mode, DSA tag, or Extended DSA Tag, the networking controller performs zero padding to packet sizes of 66, 68, or 72B, respectively. This guarantees that the packet is still legal after the destination MAC strips the extra bytes of Marvell Header or DSA tag.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

12.2.1 Address Decoding

The Mbus port that serves the networking controller has six address windows. Each address window can be individually configured.

With the networking controller DMA transaction (buffer read/write, descriptor read/write), the address is compared against the address decoding registers. Address comparison is done to select the correct target interface and attributes.

Four of the address windows have an upper 32-bit address register, for 64-bit addressing support. These are used for accessing interfaces that support more than 4-GB address space. The address generated on the interface is composed of the 32-bit address issued by the DMA (if it hits the relevant address window) concatenated with the High Address Remap register content.

To ensure that the networking controller DMA avoids accessing a forbidden address space (due to a programming bug), the Mbus port uses access protection logic that prevents it from read/write accesses to specific address windows.

If the address does not match any of the address windows, or if it violates the access protection settings, an interrupt is generated. The transaction is executed but not to the original address. Instead, the transaction is executed to a default address and target as specified in the Default Address (see Ethernet Unit Default Address (EUDA) Register (Table 530 p. 1144)) and ID registers (see Ethernet Unit Default ID (EUDID) Register (Table 531 p. 1144)).

12.2.2 Endianess and Swap Modes

Each DMA channel has configurable behavior on Little or Big Endian support, per DMA channel data receive (including virtual address read from the buffer) and data transmission. (see the <BLMR> field and the <BLMT> field in the SDMA Configuration (SDC) Register (Table 538 p. 1148).

For every DMA channel, descriptor accesses may be swapped (see the <SwapMode> field in the SDMA Configuration (SDC) Register (Table 538 p. 1148).

12.2.3 Hardware Parser Overview

The basic parser mode is defined by setting the <AccelerationMode> field in the Port Acceleration Mode (PACC) Register (Table 539 p. 1149).

Each networking controller includes in its receive path, a Layer-2 Destination Address (MAC-DA) recognition and filtering mechanism of up to 16-Unicast MAC addresses, 256 IP Multicast addresses, and 256 Multicast/Broadcast address. The networking controller may also detect Layer-2 frame-type encapsulation, as well as common Layer 3 and Layer 4 protocols.

Queue classification on received traffic is assigned to the DMA queue based upon a configurable analysis, which evaluates the DA-MAC, IP TOS (Type of Service), IEEE 802.1p priority tag, and protocol (ARP, TCP, or UDP). An example for use of this feature is the implementation of differentiated services or real-time, jitter-sensitive voice/video traffic intermixed with data traffic. As each queue has its own buffering, blocking is avoided and latency is reduced for CPU service.

Copyright © 2015 Marvell

Frame type/encapsulation detection is available on:

Layer 2 for:
- Bridge Protocol Data Unit (BPDU)
- VLAN (programmable VLAN-EtherType)
- Ethernet v2, LLC/SNAP

Layer 3 for:
- IPv4
- IPv6

Layer 4 (over IPv4/IPv6) for:
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Frame enqueueing is in accordance with DA, VLAN-802.1p, IP-TOS using the *highest* priority counts. Frame enqueueing is captured according to protocol type for TCP, UDP, ARP, or BPDU. The networking controller also supports enqueuing based on the proprietary Marvell® Header or the DSA Tag. These are useful when interfacing Marvell FE or GbE switches that utilize these features, resulting in improved routing performance, and support for cascade and stack of switches.

12.2.4 IP Checksum Generation

IPv4 checksum may be calculated during the packet DMA from memory, and it is replaced in the checksum field for IPv4 packets. One bit in the transmit descriptor is used for specifying if the IP checksum generation is required for a specific packet.

12.2.5 TCP Checksum Generation

The TCP checksum may be enabled per frame. When TCP checksum is enabled, it is calculated during the packet read from memory by the DMA and the calculated value is written in the checksum field before transmission begins.

Since a TCP segment may be transmitted over several Ethernet packets, and since the checksum in the next packets continue the checksum calculation of previous packets, there are two types of checksum generation commands (depending on the <GL4chk> field in the Tx descriptor):

- Calculate the checksum on the *first* packet in the segment:
  In this case, the 16-bit checksum field in the descriptor must be zero. The checksum is done fully by the device, and includes parsing the header according to the descriptor fields and calculating the checksum on pseudo-header. The checksum continues with full checksum calculation on the TCP data, and finally it is placed in the packet before transmission.
- Calculating checksum on *non-first* packets in the segment:
  The CPU is required to calculate the initial checksum, including the pseudo-header checksum in the <L4iChk> field in the Tx descriptor. The DMA uses this initial checksum value in calculating the TCP checksum over the TCP payload in the packet and places it in the TCP checksum field of the packet before transmission.

The CPU may choose to always calculate the checksum over the pseudo-header, and let the hardware calculate of the payload checksum.

Copyright © 2015 Marvell

12.2.6 UDP Checksum Generation

The UDP checksum generation is the same as the TCP checksum generation support, with both first and non-first modes (see Section 12.2.5, TCP Checksum Generation, on page 166).

12.2.7 Frame Type Indications

The receive processing of the frame (see Section 12.3, Queues Management – Counters Mode, on page 169) allows passing various useful indications about each packet in the Rx descriptor. These indicators include MAC level errors (like Ethernet CRC check fail), to facilitate CPU processing overhead in packet header processing, and in Layer 3 and Layer 4 checksum calculations.

See the descriptor description for details. For a definition of the indications, see Section 12.3.2.4, Receive Descriptor Structure in Hardware Parsing Mode, on page 177.

12.2.8 TCP Checksum Checking

TCP frames include a 16-bit checksum that protects the entire segment payload (that usually spans over a number of packets) as well as the TCP header and some of the IP fields.

Frames may be received in an interleaved fashion from different TCP connections, and also out of order, within any TCP connection.

The Rx frame parsing allows offloading most of the overhead from the software. The Rx descriptor (see Section 12.3.2.4, Receive Descriptor Structure in Hardware Parsing Mode, on page 177) provides frame type indications such as:

- IPv4
- IPv6
- Validity of IP header with correct IPHL, IPTL, and IPv4 checksum checked OK
- Layer 2 encapsulation information (VLAN, Ethernetv2 or LLC/SNAP)
- TCP or UDP type detection.

TCP over IPv6 checksum check results are generated in the Rx descriptor. The checksum check is supported for IPv6 packets without extension headers.

TCP over IPv4 checksum check results are generated in the Rx descriptor in the following way, where two cases are identified—fragmented or non-fragmented IP packets:

1. If it is a non-fragmented IPv4 packet (the packet's IP Header Flag <MF> = 0 and <Offset> = 0x0), meaning, the packet includes both TCP header and the full payload, (and therefore TCP checksum can be calculated), the descriptor will be closed with an indication that the frame is not fragmented. The TCP checksum calculation also takes into account the pseudo-header if the <RxCS> field in the Port Configuration (PxC) Register (Table 621 p. 1193) is set to 1. If the <RxCS> field is set, the L4 checksum compare result will be valid (descriptor's <L4ChkOK> bit is valid).



Note

The length field for the pseudo-header is taken from the following operation:
`IPTL - IPHL * 4.`

   For the checksum calculation, the value 0x0 is used instead of the checksum field in the received frame as required by the standard. In addition, the checksum calculation for each frame always starts with the initial value of 0x0.

2. If it is a fragmented IPv4 packet (either <MF> != 0 or <Offset> != 0) or if the <RxCS> field is cleared to 0 (calculation without pseudo-header), the pseudo-header is not calculated in the checksum. The checksum is calculated only on the TCP segment (header + data) and places

Copyright © 2015 Marvell

the result in the first descriptor of each frame. Therefore, the checksum compare bit (L4ChkOK bit) is not valid.

Note

For fragmented IPv4 packets, the checksum calculation does not put a zero in the checksum field, and therefore, in a frame that is the first fragment of IP packet (<Offset> =0x0 and <MF>!= 0), the checksum result can be wrong (since it includes the checksum field of the TCP header). This should be corrected by the software.

The networking controller calculates the checksum per each packet. The software should sum all the checksum calculations for the complete IP packet and compare it to the checksum field in the TCP header.

12.2.9 UDP Checksum Checking

UDP frames include a 16-bit checksum that protects the entire segment payload (that usually spans over a number of packets) as well as the UDP header and some of the IP fields.

Frames may be received in an interleaved fashion from different UDP streams, and also out of order within any UDP stream.

The Rx frame parsing allows offloading most of the overhead from the software. The Rx descriptor provides frame type indications such as:

- IPv4 validity of IP header with correct IPHL, IPTL, and IP checksum checked OK
- IPv6
- Layer 2 encapsulation info (VLAN, Ethernetv2 or LLC/SNAP)
- TCP or UDP type detection

UDP over IPv6 checksum check results are written to the Rx descriptor. The checksum check is supported for IPv6 packets without extension headers.

UDP over IPv4 checksum results are written to the Rx descriptor, when two cases are identified:

- Non-fragmented IPv4 packet
- Fragmented IPv4 packet

Non-Fragmented IPv4 Packet

If it is a non-fragmented IPv4 packet (the packet IPv4 Header Flag <MF> = 0 and <Offset> = 0x0):

- The packet includes both the UDP header and the full payload, and therefore UDP checksum can be calculated.
- The descriptor is closed with an indication that the frame is not fragmented. (The descriptor's <IPv4_Frg> bit cleared to 0).
- The L4 checksum compare results are valid.

The UDP checksum calculation also takes into account the pseudo-header, if the <RxCS> field in the Port Configuration (PxC) Register (Table 621 p. 1193) is set to 1.

Note

The length field for the pseudo-header is taken from the following operation:
`IPTL - IPHL * 4`.

For the checksum calculation, the value 0x0 is used instead of the checksum field in the received frame, as required by the standard. In addition, the checksum calculation for each frame always starts with the initial value of 0x0.

Document Classification: Proprietary Information

If the frame checksum is 0x0, the checksum function does not compare any values, and closes the descriptor as checksum OK. According to the standards, this is the indication that checksum function was disabled.

Fragmented IPv4 Packet

If it is a fragmented IPv4 packet (either <MF> != 0 or <Offset> != 0) or if the <RxCS> field is cleared to 0 (calculation without pseudo-header), the pseudo-header is not calculated in the checksum. The checksum is calculated only on the UDP segment (header + data), and places the result in the first descriptor of each frame. The checksum compare bit (L4ChkOK bit) is not valid.



Note

For fragmented IP packets, the checksum calculation does *not* put zero in the checksum field. In a frame that is the first fragment of IP packet (<Offset> =0x0 and <MF>!= 0), the checksum result can be wrong because it includes the checksum field of the UDP header. This issue should be corrected by the software.

The networking controller calculates the checksum per packet. The software should sum all the checksum calculations for the complete IP packet, and compare this value to the checksum field in the UDP header.

12.3 Queues Management – Counters Mode

This section describes the queues management functionality, when working in Enhanced (by counters) Queue Management mode.

This mode is defined by setting the <AccelerationMode> field in the Port Acceleration Mode (PACC) Register (Table 539 p. 1149) to one of the Enhanced modes.

12.3.1 Transmit DMA

This section describes the descriptors management methodology.

12.3.1.1 Descriptors Management by Counters

The descriptors of each queue are located in successive locations in a preassigned external memory space (for example, DRAM).

The Tx descriptors are 32B aligned.

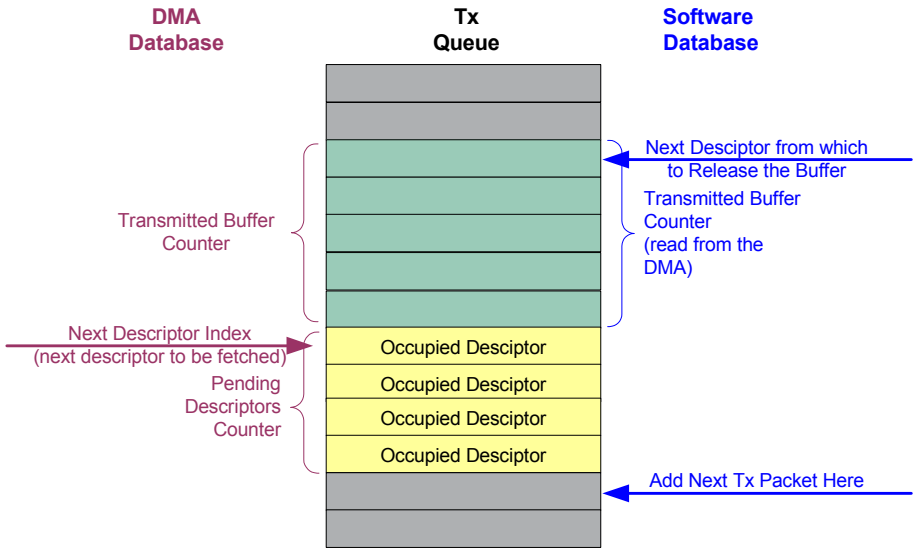
Figure 44 provides a block diagram of the transmit queue management counters.

The following parameters are maintained for each queue:

- Queue starting address, configured by the <DescriptorsQueueStartingAddress> field in the Port TX Queues Descriptors Queue Address (PTXDQA)<q> Register (q=0–7, TCONT_NO=8–8) (Table 596 p. 1181).
- Queue size, configured by the Port TX Queues Descriptors Queue Size (PTXDQS)<q> Register (q=0–7, TCONT_NO=8–8) (Table 597 p. 1181).

Copyright © 2015 Marvell

Figure 44: Descriptors Management by Counters Transmit Queue



The networking controller maintains the following counters.

Pending Descriptors Counter

Number of descriptors pointing to the buffers that contain Tx packets yet to be processed by the networking controller. This counter can be read in the <PendingDescriptorsCounter> field in the Port TX Queues Status (PTXS)<q> Register (q=0–7, TCONT_NO=8–8) (Table 598 p. 1182).

Following the preparation of new descriptors, the sender (for example, the CPU) increments this counter by accessing the <NoOfWrittenDescriptors> field in the Port Tx Queues Status Update (PTXSU)<q> Register (q=0–7, TCONT_NO=8–8) (Table 599 p. 1183).

The networking controller decrements this counter after closing the last descriptor in packet. The decrement is by the number of descriptors occupying the packet.

Transmitted Buffers Counter

Number of buffers that were transmitted, but not yet released by the software. This counter can be read in the <TransmittedBuffersCounter> field in the TX Transmitted Buffers Counter (TXTBC)<q> Register (q=0–7, TCONT_NO=8–8) (Table 601 p. 1183).

The networking controller increments this counter by the number of descriptors used for specific packet (one or more), after writing the last descriptor associated with the packet.

The CPU decrements this counter by writing to the <NoOfReleasedBuffers> field in the Port Tx Queues Status Update (PTXSU)<q> Register (q=0–7, TCONT_NO=8–8) (Table 599 p. 1182).

Next Descriptor Index

Index of the next descriptor to be fetched. This counter can be read in the <NextDescriptorIndex> field in the Port TX Queues Descriptor Index (PTXDI)<q> Register (q=0–7, TCONT_NO=8–8) (Table 600 p. 1183). This field is implemented as a cyclical counter; it counts from 0 to <DescriptorsQueueSize> minus 1.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

If there are no descriptors remaining in the queue, as indicated by the <PendingDescriptorsCounter> field in the Port TX Queues Status (PTXS)<q> Register (q=0–7, TCONT_NO=8–8) (Table 598 p. 1182), further packet transmission from this queue is disabled. When addition of new descriptors to the queue is indicated by the CPU, by writing to the <NoOfWrittenDescriptors> field in the Port Tx Queues Status Update (PTXSU)<q> Register (q=0–7, TCONT_NO=8–8) (Table 599 p. 1183), the packets transmission is resumed.



Note

If it is required to start transmission from the first descriptor in the queue, in all queues, the CPU must set the <TxDMAInit> field in the Port TX Initialization (PTXINIT) Register (Table 602 p. 1184).

12.3.1.2 Tx Data Read from Buffers

The packet data is read from the buffers starting from the offset defined in the descriptor's <PacketOffset> field.

This is done for all of the buffers, not just the first buffer.

12.3.1.3 Transmit Descriptor Structure

Table 26 shows the Transmit Descriptor structure.

Table 26: Transmit Descriptor Description

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Offset
Byte3								**Byte2**								**Byte1**								**Byte0**								
GL4chk [1:0]	Packet_offset[6:0]							buf_mod F	L	Padding	GlPchk	L3Type	L4Type	Reserved	Pool_ ind	IP_HdLen [6:2]				Reserved	L3_offset[6:0]											+0
Byte Count[15:0]																L4iChk[15:0]																+4
Buffer Pointer (physical address) [31:0]																																+8
DSA_Tag[1:0]	Reserved											Reserved								MH_ select [3:0]				Reserved							ES	+C
Reserved																																+10
Reserved																																+14
Reserved																																+18
Reserved																																+1C

Table 27 through Table 34 lists the transmit descriptor fields, based on the offset in use.

Table 27: Transmit Descriptor—Offset 0

Bits	Name	Description
6:0	L3_offset	Layer 3 Offset The offset of Layer 3 header beginning (in bytes), from the beginning of the packet. It is used for checksum calculation. **NOTE:** This field is valid only for the first descriptor of a packet (F==1).
7	Reserved	Reserved
12:8	IP_HdLen	IP Header Length Provides the length in long words (4B) of the IP header. Supports up to 32 long words =128B. **NOTE:** This is only valid if the F bit [21] is set, and either L4 or IPV4 checksum generation are required by <GL4chk> field (bits [31:30]) and <GIPchk> field (bit 18).
14:13	Pool_ind	Pool Index This field is only valid if buf_mod field in the descriptor = 1 (buffer released by the BM unit), and the BM unit is used, as selected by the acc_mode of the PxACC register. It contains the pool index, to which the buffer should be released.
15	Reserved	Reserved
16	L4Type	Layer 4 Type 0 = TCP 1 = UDP When Layer 4 checksum generation is required, according to <GL4chk> field, then this bit indicates which Layer 4 protocol is carried in the frame. **NOTE:** This is only valid if the <F> bit[21] is set.
17	L3Type	Layer 3 Type 0 = IPv4 1 = IPv6 **NOTE:** This bit is only valid if either IPv4 header checksum or L4 checksum are to be generated (according to <GIPchk> and <GL4chk> fields.
18	GIPchk	Generate IPv4 checksum. 1 = Generate IPv4 header checksum 0 = Do not generate IPv4 header checksum **NOTE:** This is only valid if the <F> bit[21] is set and L3Type indicates IPV4 packet.
19	Padding	Padding When this bit is set and the packet is smaller than 60B, zero-value bytes are appended to the packet. Use this feature to prevent transmission of fragments. **NOTE:** This is only valid if <F> bit[21] is set. If disabling hardware CRC generation (i.e. CRC is generated by the software), zero padding must be disabled.(Table 539 p. 1149)
20	L	Last When equal to 0x1, indicates the last buffer used for the frame.
21	F	First When equal to 0x1, indicates the first buffer used for the frame.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 27: Transmit Descriptor—Offset 0 (Continued)

Bits	Name	Description
22	Buf_mod	Buffer Release Mode 0 = The buffer should be release by software. 1 = The buffer should be released by the networking controller (to the Buffer Management unit). **NOTE:** This bit is valid for all descriptors of a packet (not only for the first).
29:23	Packet_offset	Packet Offset The offset of the packet (in bytes) from the beginning of the buffer. **NOTE:** This offset is valid for all buffers (not only for the first).
31:30	GL4chk	Generate Layer 4 (TCP/UDP) Checksum 0 = Generate L4 checksum. Packet is fragmented and this is the first fragment. The CPU provides in the <L4iChk> field of the descriptor the checksum calculated over the pseudo header of the first fragment and over the TCP/UDP payload of the second to last fragments. The networking controller calculates the checksum over TCP/UDP header + L4 payload (starting from L3 offset + L3 header length * 4), adds to that the initial checksum value supplied by <L4iChk> field in Tx descriptor and inserts the calculated checksum to relevant place in UDP/TCP header. 1 = Generate L4 checksum. Packet is not fragmented. The networking controller Generates L4 checksum as required by UDP/TCP standard over IPv4/IPv6 pseudo header + TCP/UDP header + L4 payload, and inserts checksum to the UDP/TCP header. 2 = Do not generate L4 checksum 3 = Reserved The payload length over which the checksum is calculated is determined as: **For IPV4 packets:** “Total Length” field in the IPV4 header - 4*IP_HdLen(5b) field in the descriptor. **For IPV6 packets:** The “Payload Length” field in the IPV6 header. **NOTE:** This field is only valid if the <F> bit[21] is set.

Table 28: Transmit Descriptor—Offset 4

Bits	Name	Description
15:0	L4iChk	Layer 4 Initial Checksum In this field, the CPU provides the initial checksum value calculated on the pseudo header and the payload of the second to last fragments of a fragmented TCP/UDP segment. It is valid only for the first fragment of a fragmented TCP/UDP segment (indicated by <GL4ichk> == 0). **NOTE:** Only valid if the <F> bit[21] is set.
31:16	Byte Count	Byte Count Number of bytes to be transmitted from the associated buffer. This is the payload size in bytes.

Copyright © 2015 Marvell

Table 29: Transmit Descriptor—Offset 8

Bits	Name	Description
31:0	Buffer_physical _pointer	A 32-bit pointer to the beginning of the buffer associated with this descriptor.

Table 30: Transmit Descriptor—Offset C

Bits	Name	Description
0	ES	Error Summary This field is written by the networking controller. Error Summary of MAC level errors on frame transmission. 0 = No Error 1 = Error occurred (Late Collision, or Retransmit Limit, or Underrun Error) **NOTE:** This field is only valid if <L> bit[20] is set.
2:1	EC	MAC Error Coding This field is written by the networking controller. 00 = LC (Late Collision) 01 = UR (Underrun) 10 = RL (retransmit limit) reached (excessive collision) 11 = Reserved **NOTE:** Valid only if <L> bit[20] is set and <ES> bit[0] is set.
3	Reserved	Reserved
7:4	MH_ select	Marvell Header select. This field defines how the generated Tx packet header (the first 2B of the packet) is created: 0 = No change of the header. packet is transmitted with same header as in the buffer (may be used when the packet is received from CPU). 1 = Replace the header with the value of TX_MH_reg_1 configuration field. 2 = Replace the packet header with the value TX_MH_reg_2 configuration contents. 3 = Replace the packet header with the value TX_MH_reg_3 configuration contents. 4 = Replace the packet header with the value TX_MH_reg_4 configuration contents. 5 = Replace the packet header with the value TX_MH_reg_5 configuration contents. 6 = Replace the packet header with the value TX_MH_reg_6 configuration contents. 7 = Replace the packet header with the value TX_MH_reg_7 configuration contents. 8 = Replace the packet header with the value TX_MH_reg_8 configuration contents. 9 = Replace the packet header with the value TX_MH_reg_9 configuration contents. 10 = Replace the packet header with the value TX_MH_reg_10 configuration contents. 11 = Replace the packet header with the value TX_MH_reg_11 configuration contents. 12 = Replace the packet header with the value TX_MH_reg_12 configuration contents. 13 = Replace the packet header with the value TX_MH_reg_13 configuration contents. 14 = Replace the packet header with the value TX_MH_reg_14 configuration contents. 15 = Replace the packet header with the value TX_MH_reg_15 configuration contents. **NOTE:** The packet header change is performed only if Marvell Header is enabled, by the MHEn bit in the legacy Marvell Header register.
19:8	Reserved	Reserved
29:20	Reserved	Reserved

Copyright © 2015 Marvell

Table 30: Transmit Descriptor—Offset C (Continued)

Bits	Name	Description
31:30	DSA_Tag	DSA Tag Indicates the type of DSA tag in the packet: 0 - no DSA tag 1 - 1 word DSA tag 2 - 2 words DSA tag 3 - Reserved

Table 31: Transmit Descriptor—Offset 10

Bits	Name	Description
31:0	Reserved	Reserved

Table 32: Transmit Descriptor—Offset 14

Bits	Name	Description
31:0	Reserved	Reserved

Table 33: Transmit Descriptor—Offset 18

Bits	Name	Description
31:0	Reserved	Reserved

Table 34: Transmit Descriptor—Offset 1C

Bits	Name	Description
31:0	Reserved	Reserved

12.3.2 Receive DMA

This section describes the descriptors management methodology in the context of Rx DMA.

12.3.2.1 Queues Management by Counters

The descriptors for each queue are located in successive locations in a pre-assigned descriptor queue.

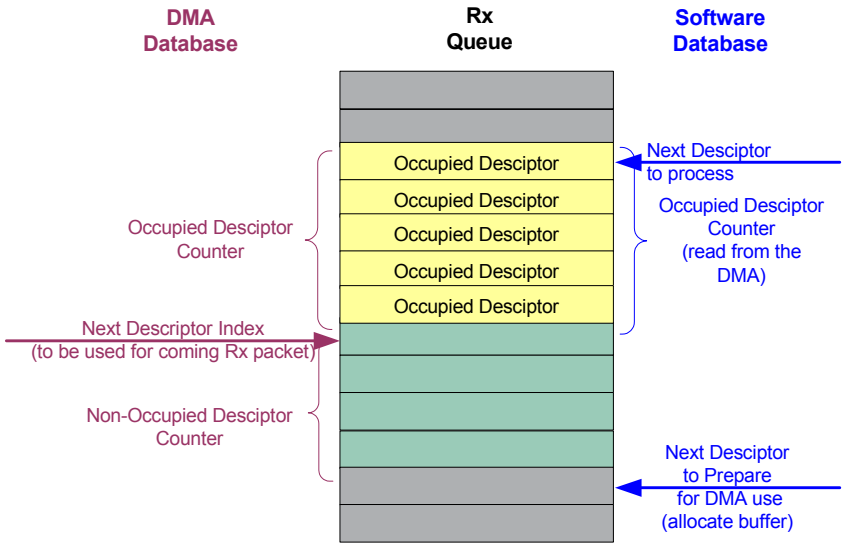
The queue starting address is configured by the <DescriptorsQueueStartingAddress> field in the Port RX Queues Descriptors Queue Address (PRXDQA) <q> Register (q=0–7) (Table 566 p. 1164).

The queue size is configured by the <DescriptorsQueueSize> field in the Port RX Queues Descriptors Queue Size (PRXDQS)<q> Register (q=0–7) (Table 567 p. 1165).

Figure 45 provides a block diagram of the receive queue management counters.

Copyright © 2015 Marvell

Figure 45: Descriptors Management by Counters Receive Queue



The networking controller maintains the following counters for each Rx queue.

Occupied Descriptors Counter

This counter indicates the number of descriptors pointing to the buffers that contain Rx packets, and are not yet processed by the CPU. It can be read in the <OccupiedDescriptorsCounter> field in the Port RX Queues Status (PRXS)<q> Register (q=0–7) (Table 569 p. 1167).

The networking controller increments this counter following a packet data transfer to the buffer.

The CPU decrements this counter by writing to the <NoOfProcessedDescriptors> field in the Port RX Queues Status Update (PRXSU)<q> Register (q=0–7) (Table 570 p. 1167).

Non-Occupied Descriptors Counter

This counter indicates the number of descriptors that are received from the CPU and are unused by the networking controller. It is read in the <NonOccupiedDescriptorsCounter> field in the Port RX Queues Status (PRXS)<q> Register (q=0–7) (Table 569 p. 1166).

The CPU increments this counter by writing to the <NoOfNewDescriptors> field in the Port RX Queues Status Update (PRXSU)<q> Register (q=0–7) (Table 570 p. 1167).

The networking controller decrements this counter after writing to the descriptor.

Next Descriptor Index

This counter indicates the index of the next descriptor to be fetched. It is read in the <NextDescriptorIndex> field in the Port RX Descriptor Index (PRXDI)<q> Register (q=0–7) (Table 571 p. 1168).

This is a cyclical counter that counts from 0 to <DescriptorsQueueSize>DescriptorsQueueSize, minus1.

Copyright © 2015 Marvell

Note If it is required to start reception from the first descriptor in the queue, for all queues, the CPU must set the <RXDMAInit> field in the Port RX Initialization (PRXINIT) Register (Table 576 p. 1170).

12.3.2.2 Rx Data Transfer to Buffers

This section describes the Rx data transfer to buffers, when queue management mode is Enhanced Mode (Queues management by counters).
When the received packet is written to the Rx buffer the data is written as follows:

- For the first buffer:
  - The packet is written starting from the <PacketOffset> field in the Port RX Queues Configuration (PRXC)<q> Register (q=0–7) (Table 564 p. 1163) plus 2B.
  - If a Marvell® Header (2B length) exists, it is written to the buffer starting from <PacketOffset>. Otherwise, 2B of zeros are written instead of the Marvell Header.
  - The offset configured is according to the <PacketOffset>.
- For the second to last buffer:
  - If the buffer is allocated by the software, the packet is written from the first byte of the packet.
  - If the buffer is allocated by the BM unit, the packet is written to the buffer, starting from an offset of 8B. This is done so the virtual buffer address is not overwritten. The virtual buffer address is written in the first 4B of the buffer.

12.3.2.3 Rx Buffer Virtual Pointer

Note Available when Buffer Management unit is enabled.

Besides the physical address, each buffer has a virtual address that is used by the CPU to access the buffer.

The CPU must store each buffer's virtual address in the first Dword (4B) of the buffer.

During the DMA, the networking controller writes the packet data with an offset, as described in Section 12.3.2.2, so that the virtual address is not overwritten.

When writing to the descriptor, the networking controller copies the virtual address from the buffer's first Dword to the buffer_virtual_pointer field of the descriptor (see Table 35, Receive Descriptor – Description, on page 178).

12.3.2.4 Receive Descriptor Structure in Hardware Parsing Mode

This section describes the receive descriptor structure, in hardware parsing mode, as defined by the <AccelerationMode> field in the Port Acceleration Mode (PACC) Register (Table 539 p. 1149)— when it is set to 0x1 or 0x2.

- The descriptor length is 32B, and it must be 32B aligned (that is, Descriptor_Address[4:0] ==00000).
- Descriptors can reside anywhere in the address space.
- Receive buffers associated with receive descriptors are limited to 64-1 KB and must be 64-bit aligned (that is, Buffer_Address[2:0]==000).
- The minimum buffer size for the receive buffer is 64B.

Copyright © 2015 Marvell

Table 35 shows the receive descriptor structure.

Table 35: Receive Descriptor – Description

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Offset
Byte3								Byte2								Byte1								Byte0								
IPv4_Frg	L4ChkOK	Reserved	U	F	L	IPHeadOK	L3IP	Layer2Ev2	Layer4[1:0]		BPDU	VLAN	EC		ES	Reserved	Pool_ind[1:0]		Reserved													+0
Byte Count[15:0]																Reserved																+4
Buffer_physical_pointer[31:3]																														Reserved		+8
Reserved																																+C
Buffer_virtual_pointer[31:0]																																+10
L4Chk[15:0]																Reserved																+14
Reserved																															Reserved	+18
Reserved																																+1C

Table 36 through Table 42 lists the receive descriptor fields, based on the offset in use.

Table 36: Receive Descriptor—Offset 0

Bits	Name	Description
12:0	Reserved	
14:13	Pool_ind	Pool index to which the buffer belongs. This field is valid only when the Buffer Management unit is enabled by the networking controller (as configured in the acc_mode field of the PxACC register), and is enabled for the queue (as configured in <HwBuffAloc> field in the PxRXyC register.
15	Reserved	Cleared to 0 by the controller.
16	ES	Error Summary 0 = No Error 1 = Error Occurred (CRC Error, Overrun Error, Maximum Frame Length Error, Resource Error) **NOTE:** This is only valid if <F> bit[27] is set.

Copyright © 2015 Marvell

Table 36: Receive Descriptor—Offset 0 (Continued)

Bits	Name	Description
18:17	EC	MAC Error Coding 00 = CE - CRC Error 01 = OR - Overrun Error 10 = MF - Maximum Frame Length Error. Frame is longer than the MAX_FRAME_SIZE. 11 = RE - Resource Error (No descriptors in the middle of the frame) **NOTE:** This is only valid if the <F> bit[27] and the <ES> bit[16] are set. If multiple errors occurred, then the reporting priority is: Resource Error, Maximum Frame Length Error, Overrun Error, and CRC Error.
19	VLAN	When equal to 0x1, VLAN Frame is VLAN tagged (according to programmed VLAN-EtherType). **NOTE:** This is only valid if the <F> bit[27] is set, and the <ES> bit[16] is cleared to 0.
20	BPDU	Bridge Protocol Data Unit Set to 0x1 when the frame is BPDU. **NOTE:** Only valid if the <F> bit[27] and <ES> bit[16] are cleared to 0.
22:21	Layer4	Frame encapsulation and protocol. 00 = Frame is TCP over IPv4/IPv6 over Ethernet v2 or LLC/Snap (with or without VLAN tag). The Checksum result is provided in the <L4Chk> bits[18:3]. 01 = Frame is UDP over IPv4/IPv6 over Ethernet v2 or LLC/Snap (with or without VLAN tag). The Checksum result is provided in the <L4Chk> bits[18:3]. 10 = Other Frame type. 11 = Reserved **NOTE:** TCP/UDP parsing over IPv6 is supported for IPv6 header without the extension header. This is only valid if the <F> bit[27] is set, and the <ES> bit[0] is cleared to 0. This is only valid if the <IPHeadOK> bit[25] is set and <L3IP> bit[24] is set.
23	Layer2Ev2	Set if Layer 2 is Ethernet v2. **NOTE:** This is only valid if the <F> bit[27] is set, and the <ES> bit[16] is cleared to 0.
24	L3IP	• <L3IP> = 0, <IPHeadOK> = 0: Unknown L3 protocol (was not recognize as IPv4 or IPv6). • <L3IP> = 0, <IPHeadOK> = 1: L3 type is IPv6. (IP header <version> field = 6 and EtherType = 0x86DD over Ethernet v2, or over LLC/SNAP, with or without VLAN tag). • <L3IP> = 1, <IPHeadOK> = 0: L3 type is IPv4. (EtherType = 0x800 over Ethernet v2, or over LLC/SNAP, with or without VLAN tag). IP header is not OK (OK as defined below). • <L3IP> = 1, <IPHeadOK> = 1: L3 type is IPv4. (EtherType = 0x800 over Ethernet v2, or over LLC/SNAP, with or without VLAN tag). IP header is OK (OK as defined below). IPv4 header is OK if all the following occur: • IP header <version> field = 4) • IPHL >=5 • IPHL *4<= IPTL • IPv4 header Checksum is OK. **NOTE:** This is only valid if the <F> bit[27] is set, and the <ES> bit[0] is cleared to 0.
25	IPHeadOK	
26	L	Last When equal to 0x1, indicates the last buffer used for the frame.
27	F	First When equal to 0x1, indicates the first buffer used for the frame.

Table 36: Receive Descriptor—Offset 0 (Continued)

Bits	Name	Description
28	U	Unknown Destination Address The frame is Unicast and was not matched to the MAC Address Base (DA[47:4]). **NOTE:** This is only set if working in promiscuous mode. See <UPM> field in the Port Configuration (PxC) Register (Table 621 p. 1195). This is only valid if the <F> bit[27] is set, and the <ES> bit[16] is cleared to 0.
29	Reserved	Cleared to 0 by the controller.
30	L4ChkOK	Layer4 Checksum OK 1 = OK (passed). 0 = Check failed. If <Layer4> bits[22:21] is 01(UDP indication) and the received frame checksum field was 0x0, then the bit will indicate passed. This is only valid if the <IPHeadOK> bit[25] is set, and <L3IP> bit[24] is set. This is only valid if <Layer4> field is 00 or 01.(indicating TCP/UDP). This is only valid if the <F> bit[27] is set, and the <ES> bit[0] is cleared to 0. For an IPv4 packet, this is only valid if the Receive Descriptor <IPv4_Frg> field is cleared (That indicates a non-fragmented IPv4 packet). This is only valid if <RxCS> field in the Port Configuration (PxC) Register (Table 621 p. 1193) is set to 1 (indication for L4 calculation including pseudo header).
31	IPv4_Frg	IPv4 fragmented indication. 1 = Fragmented 0 = Not fragmented This is only valid if the <IPHeadOK> bit[25] and the <L3IP> bit[24] are set. This is only valid if the Receive Descriptor Command/Status Register <F> bit[27] is set, and <ES> bit[0] is cleared to 0.

Table 37: Receive Descriptor—Offset 4

Bits	Name	Description
15:0	Reserved	Reserved
31:16	Byte Count	When a descriptor is closed, this field is written by the DMA, indicating the number of bytes actually written by the DMA into the buffers. **NOTE:** This is only valid if the <F> bit[27] is set.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 38: Receive Descriptor—Offset 8

Bits	Name	Description
2:0	Reserved	Cleared to 0 by the controller.
31:3	Buffer_physical_pointer	Written by DMA or CPU The 29 MSB of a 32-bit physical pointer to the beginning of the buffer associated with this descriptor. • When the queue is configured as software buffer allocation (see <HwBuffAloc> field in the PxRXyC register), then this field is written by the CPU. • When the queue is configured as a hardware buffer allocation (see <HwBuffAloc> field in the PxRXyC register), then the Gigabit Ethernet unit writes the physical buffer pointer received from BM unit to this field. **NOTE:** The physical buffer pointer must be 64-bit aligned; therefore, bits[2:0] of the pointers are considered as zeros.

Table 39: Receive Descriptor—Offset C

Bits	Name	Description
31:0	Reserved	Reserved

Table 40: Receive Descriptor—Offset 10

Bits	Name	Description
31:0	Buffer_virtual_pointer	Buffer virtual address. This field is valid only when BM unit is used, as selected by the acc_mode field of the PxACC register and <HwBuffAloc> field in the PxRXyC register. The buffer virtual address is copied by the networking controller from the first Dword of the buffer to the descriptor.

Copyright © 2015 Marvell

Table 41: Receive Descriptor—Offset 14

Bits	Name	Description
15:0	Reserved	Reserved
31:16	L4Chk	Calculated TCP/UDP over IPv4/IPv6 Checksum **NOTE:** This is only valid if <Layer4> bits[22:21] are set to 00 or 01 (Indicating TCP/UDP). This is only valid: - If the <F> bit[27] is set and no MAC errors occurred. - If the <IPHeadOK> bit[25] is set (indicating IPv6 header or legal IPv4 header). The calculation does not include the pseudo-header if the <RxCS> field in the Port Configuration (PxC) Register (Table 621 p. 1193) is cleared to 0 (calculation without the pseudo-header). The calculation does not include the pseudo-header if <L3IP> is set (the packet is IPv4) and the Receive Descriptor <IPv4_Frg> field is set (calculation without the pseudo-header). The field equals 0x0 if the calculated UDP value is 0xFFFF.

Table 42: Receive Descriptor—Offset 18

Bits	Name	Description
31:0	Reserved	Reserved

Copyright © 2015 Marvell

13 1G/2.5G Ethernet MAC

This section describes the 1G/2.5G Ethernet MAC.

13.1 Features

The Ethernet MAC provides the following features:

- IEEE 802.3 compliant MAC layer function
- 1000/2500 Mbps operation—full duplex
- 10/100 Mbps operation—half and full duplex
- EEE—Energy Efficient Ethernet – only supported in SGMII modes
- MII/RGMII/SGMII mode
- Transmit functions:
  - Programmable values for Inter Packet Gap
  - CRC generation
  - Backoff algorithm execution
  - Error reporting
- Receive functions:
  - CRC check
  - Error report

13.2 Functional Description

The port speed, duplex, and IEEE 802.3 Flow Control can be auto-negotiated, according to IEEE standard 802.3. Backpressure is supported for half-duplex mode when operating at 10/100 Mbps speeds. Each port supports MIB counters.

Layer-2 CRC is always checked on received packets, and can be generated for transmit. This capability increases performance significantly by off-loading these operations from the software.

The port's speed (10, 100, or 1000 Mbps) is auto-negotiated through the PHY and does not require user intervention. Auto-negotiation is according to IEEE standard 802.3. The 1000 Mbps speed operates only in full-duplex mode. The 100 Mbps and 10 Mbps speeds operate either in half- or full-duplex mode, with the selection of the duplex mode auto-negotiated through the PHY without user intervention. With the RGMII interface, the port only supports symmetric Flow Control.



Note Auto-Negotiation can also be disabled. When Auto-Negotiation is disabled, the link parameters (link speed, FC, and Duplex mode) are forced by the software. The link must be forced down when changing the port speed.

Copyright © 2015 Marvell

13.3 Serial Management Interface (SMI)—Ethernet MAC

The port MAC contains a Serial Management Interface (SMI) for working with an external networking device, such as GbE PHYs.

The SMI allows control and status parameters to be passed between the device and the PHY (parameters specified by the CPU) using one serial pin (MDIO) and a clocking pin (MDC), reducing the number of control pins required for PHY mode control. Typically, the device continuously queries the PHY device for the link status, without CPU intervention. The PHY addresses for the link query are programmable in the PHY Address Register (Table 528 p. 1143).

A CPU connected to the device can write/read to/from all PHY addresses/registers. The SMI allows the CPU to have direct control over an MII-compatible PHY device via the SMI Register (Table 529 p. 1143). This control allows the driver software to place the PHY in specific modes such as Full-Duplex, Power-Down, or 1000-speed selection. It also helps control the PHY device's Auto-Negotiation function, if it exists. The CPU writes commands to the SMI register and the device reads or writes control/status parameters to the PHY device via a serial, bi-directional data pin called MDIO. These serial data transfers are clocked by the device MDC clock output.

Link Detection and Link Detection Bypass (ForceLinkPass)

Typically, the device continuously queries the PHY device for its link status, without CPU intervention. The PHY address used for the link query is determined by the PHY Address Register register, and it is programmable, where the default value is 8 for Port0, 9 for Port1. The device reads register 1 from PHY and updates the internal link bits according to the value of bit[2] of register 1. In the case of "link is down" (bit[2] is 0), the port enters the link test fail state. In this state, all of the port's logic is reset. The port exits the link test fail state only when the "link is up", bit[2] of register 1 is read from the port's PHY as 1.

The device offers the option to disable the link detection mechanism by forcing the link state of the interface to the link test pass state. This is done by forcing the register bit, and then the link status of the port remains in the "link is up" state regardless of the Interface-PHY's link bit value. The link status of the Interface-PHY can be read through the SMI from the PHY devices (register 1, bit[2]).

13.4 Port MIB Counters

The device incorporates a set of management counters, the MAC MIB Counters. They provide the necessary counters that support MAU, IEEE 802.3, and EtherLike MIB. Each port has a set of counters that reside in consecutive address spaces. Some counters are 64-bits wide.

The counters are meant to provide management software to support:

- IEEE 802.3 DTE Management objects
- Ethernet-like interface MIB: RFC 2665
- Interface MIB: RFC 2863
- Remote Network Monitoring (RMON) groups 1-4: RFC 2819



Note

The MAC MIB counters are not intended to be used for Bridge MIB nor for SMON MIB.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 43 summarizes the terms used in the definition of the counters.

Table 43: Definitions for MAC MIB Counters

Term	Definition
Collision Event	A collision has been detected before 576-bit times into the transmitted packet after Px_TXEN is asserted. Relevant to 10 Mbps and 100 Mbps speeds in half-duplex mode only.
Late Collision Event	A collision has been detected after 576-bit times into the transmitted packet after Px_TXEN. Relevant to 10 Mbps and 100 Mbps speeds in half-duplex mode only.
Excessive Collision Event	When a packet to be transmitted suffers from 15 consecutive collision events, therefore, it ought to be dropped according to the IEEE 802.3 standard. Relevant to 10-Mbps and 100-Mbps speeds in half-duplex mode only.
MRU	Maximal Receive Unit: A programmable parameter that sets the maximal length of a valid received packet.
Rx Error Event	The Receive Error signal/symbol was asserted while a frame is received.
CRC Error Event	This event occurs whenever an Ethernet frame is received and the following conditions are satisfied: 1. Packet data length is between the Minimum Frame Size - and the MRU byte size inclusive (that is, it is a valid packet data length according to the IEEE standard). 2. Packet has an invalid CRC. 3. Collision Event has not been detected. 4. Late Collision Event has not been detected. 5. Rx Error Event has not been detected.
Undersize packet	An Ethernet frame satisfying *all* of the following conditions: 1. Packet length is less than Minimum Frame Size bytes. 2. Collision Event has not been detected. 3. Rx Error Event has not been detected. 4. Packet has a valid CRC.
Fragment	An Ethernet frame satisfying *all* of the following conditions: 1. Packet data length is less than 64 bytes, OR a packet without a Start Frame Delimiter (SFD) and the packet is less than 64 bytes in length. 2. Collision Event has not been detected. 3. Rx Error Event has not been detected. 4. Packet has an invalid CRC.
Oversize packet	An Ethernet frame satisfying *all* of the following conditions: 1. Packet length is more than the MRU byte size. 2. Collision Event has not been detected. 3. Late Collision Event has not been detected. 4. Rx Error Event has not been detected. 5. Packet has a valid CRC.
Jabber	An Ethernet frame satisfying *all* of the following conditions: 1. Packet data length is greater than the MRU. 2. Packet has an invalid CRC. 3. Rx Error Event has not been detected.
Tx Error Event	An internal error event in the transmit MAC. This is a very rare situation and when it happens, it means that there the system is misconfigured.

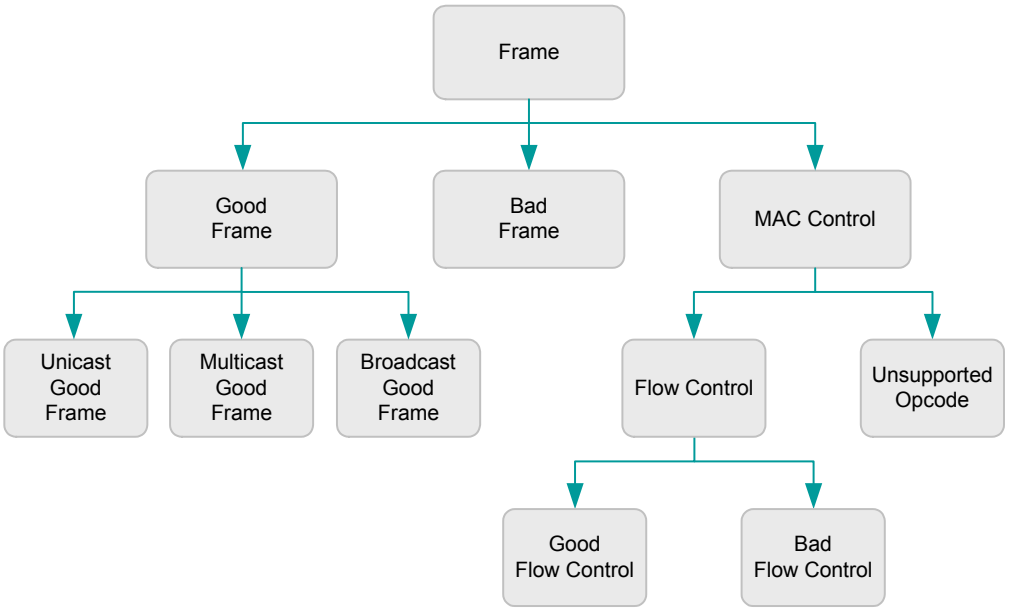
Copyright © 2015 Marvell

Table 43: Definitions for MAC MIB Counters (Continued)

Term	Definition
Bad frame	An Ethernet frame that has one of the following conditions met: CRC Error Event, Undersize, Oversize, Fragments, Jabber, Rx Error event and Tx Error Event.
MAC Control Frame	An Ethernet frame that is not a bad frame and has a value of 88-08 in the EtherType/Length field.
Flow Control Frame	A MAC Control Frame with an opcode equal to 00-01.
Good Flow Control Frame	A Flow Control frame with: 1. MAC Destination equal to 01-80-C2-00-00-01 2. 64-byte length
Bad Flow Control Frame	All Flow Control frames that are not good Flow Control frames
Good frame	An Ethernet frame that is not a bad frame NOR a MAC Control frame

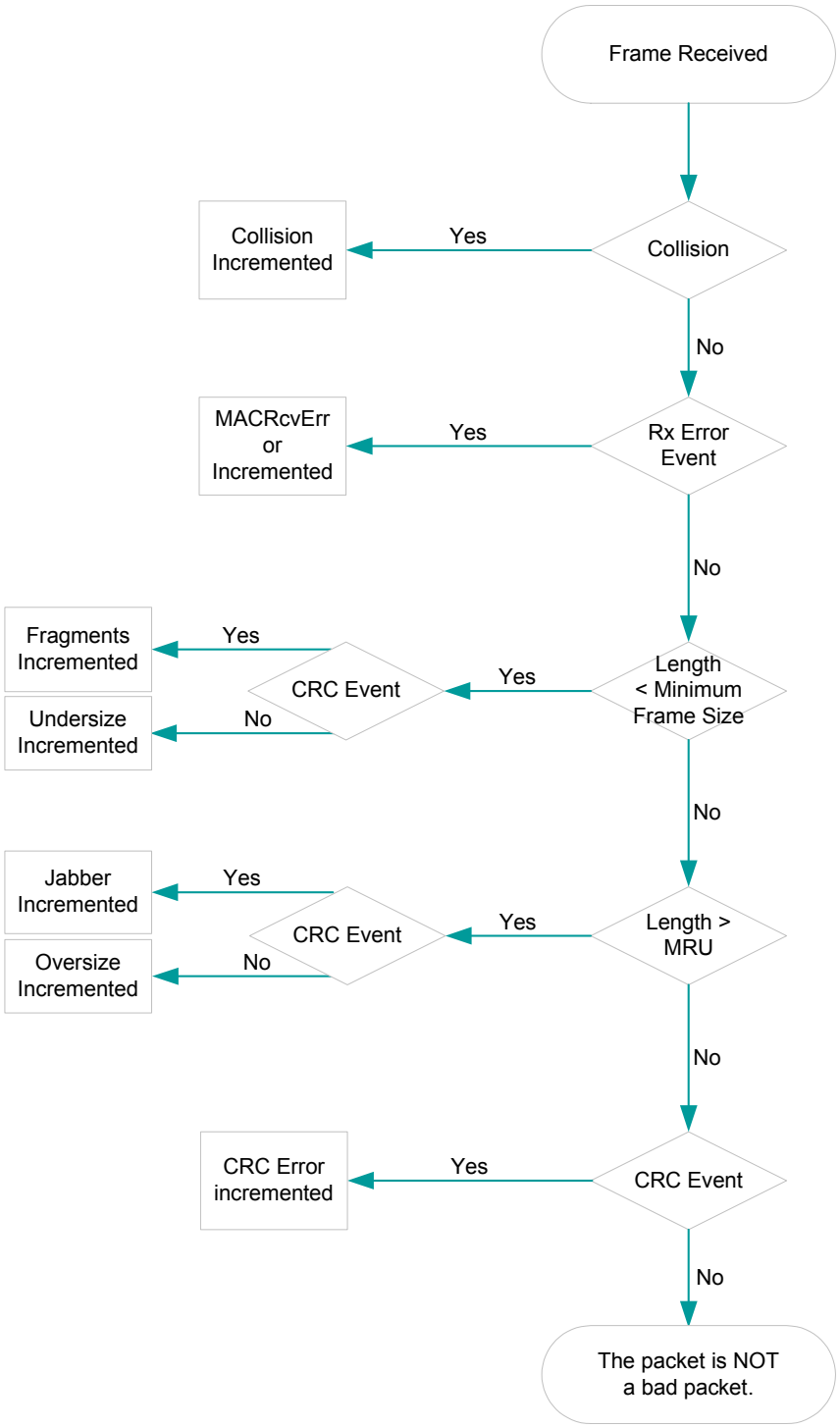
Figure 46 and Figure 47 illustrate the terms defined above.

Figure 46: Ethernet Frame Classification



Copyright © 2015 Marvell

Figure 47: Bad Frame Procedure



The counters initialize to 0 immediately after reset. Most counters are 32-bits. However, the Good Bytes received and Bytes Sent are 64-bit counters that should be read in two separate cycles.

The 64-bit counters should be read from the low address and the next read from the MIB counters must be from the high address of the same counter. The MIB interface will always return the high counter data on a read from the MIBs following a read from the low address of a 64-bit counter.

In addition to the per port counters, there are additional counters that count filtered frames for example, for MAC address lookup results. These counters are in the Port Rx Discard Frame Counter (PxDFC) Register (i=1–2) (Table 768 p. 1302) and the Port Overrun Frame Counter (PxOFC) Register (i=1–2) (Table 769 p. 1302). In conjunction with the counters block, they provide the information on total frames received.

Copyright © 2015 Marvell

14 Precise Time Protocol (PTP)

This section describes the Precise Time Protocol (PTP) Core hardware functionality.

The Precise Time Protocol defines a method for transporting the time-of-day information across a network of devices supporting this protocol. There are several industrial, consumer, and enterprise applications for PTP. For example, in consumer space, IEEE 802.1 AVB (Audio Video Bridging) defines the usage of PTP for transporting reliable audio video content across compliant equipment. PTP adds time information to nodes connected via Ethernet. This is in contrast to the IEEE 802 standard Ethernet that is an asynchronous interface where end stations do not operate using a common time base and neither do they have a concept of time.

The PTP allows multiple nodes within a given network to have a common notion of the time-of-day and also to be able to compute frequency offsets with respect to a master clock in the network.

There are 3 types of PTP frames that are supported in the device:

- IEEE 802.1AS frames on Layer 2 Ethernet
- IEEE 1588 frames on Layer 2 Ethernet
- IEEE 1588 frames on Layer 4 UDP

For IEEE 802.1AS and IEEE 1588 over Layer 2, the detection mechanism uses a special EtherType. For IEEE 1588 over Layer 4 UDP, the detection mechanism operates by checking the UDP destination port value. Once a PTP frame has been detected, event messages need to be time stamped in hardware. However, all PTP frames need to be forwarded to the CPU for further processing. Once the ingress PTP frames have been processed by the CPU, certain types of PTP frames get forwarded to other ports that again need to be detected and time stamped in hardware.

The VLAN tag of the incoming PTP frame is detected by hardware.

Using a clock selection algorithm, the PTP firmware computes the frequency and phase offset values, with respect to the best clock available in the network (labeled as the *grand master*). The computed offsets can optionally be used to control the hardware clock and time-of-day counter.

The types of frames (PTP events) that required time stamping by the hardware is configurable.

For every time-stamped frame, the sequence number from the PTP common header is captured along with the time stamp, in hardware for easier protocol software correlation.

Both IPv4 and IPv6 are supported by PTP, however, PTP does not support IPv4/IPv6 options.

Copyright © 2015 Marvell

15 Hardware Buffer Management Controller

This section describes the Hardware Buffer Management (BM) controller.

The BM performs allocation and buffer release to offload the CPU from software-intensive dynamic memory allocation and to obtain greater flexibility in allocating buffers to hardware units. The BM has four Buffer Pointer (BP) pools. Fetching or releasing a buffer from/to a pool is accomplished using read/write operations. The BM is a target on the Mbus and fetch/release operations are implemented as read/write transactions to the BM external address space. Units that access the BM controller have to allocate one of the address space windows to the BM controller's base address and target ID. The BM external address space must be configured as non-bufferable and non-cacheable.

The BM and buffer queues are initialized by the host CPU when buffers are allocated. Multiple pools are useful for supporting multiple CPUs and variable buffer sizes.

Free BPs can be allocated by the BM to the CPU or hardware units (such as the networking controller), and similarly freed buffer pointers can be released to the BM by the CPU or hardware units.

For example, during a packet forwarding application, a free BP is allocated to the GbE controller receiver for storing an incoming packet. After the packet has been forwarded, the buffer is released by the GbE controller transmitter without software buffer management overhead. During packet termination, a free buffer can be allocated to the GbE controller receiver and released by software.

The BP pools reside in external memory buffers (called BPPE) and the BM controller maintains an internal buffer pointer memory (called BPPI), to achieve low latency in allocation and release.

15.1 Features

The Hardware Buffer Management controller provides these features:

- Four separated BP pools that are managed independently. This allows for 4 different buffer and pool types or sizes.
- Up to 64-KB (16K BP) pool size (per pool).
- Dedicated internal memory of 256 BPs (per pool).
- Allocating and releasing BPs upon demand, via read and write Mbus transactions (The BM is an Mbus target).
- Up to 32 BPs can be allocated or released in a single transaction.
- Error reporting (interrupt) on a buffer pool overflow or underflow event.
- Dedicated DMA to copy BPs from external memory to internal memory and vise versa.
- 32 or 8 BPs are copied to/from external memory in a single DMA transaction.

Document Classification: Proprietary Information



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU
System on Chip

Functional Specifications – Unrestricted

Part 5: External Interfaces

- Section 16, Time Division Multiplexing (TDM) Controller
- Section 17, PCI Express Interface (PCIe) 2.0
- Section 18, Universal Serial Bus (USB) Controller
- Section 19, Serial Advanced Technology Attachment (SATA) 3.0 Host Controller
- Section 20, Audio (I^2S / S/PDIF) Interface
- Section 21, Serial Peripheral Interface (SPI)
- Section 22, Secure Digital Input/Output (SDIO) 3.0
- Section 23, Inter-Integrated Circuit Interface (I^2C)
- Section 24, Universal Asynchronous Receiver/Transmitter (UART) Interface
- Section 25, General Purpose Input/Output Ports

Doc. No. MV-S109094-U0, Rev. A

November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

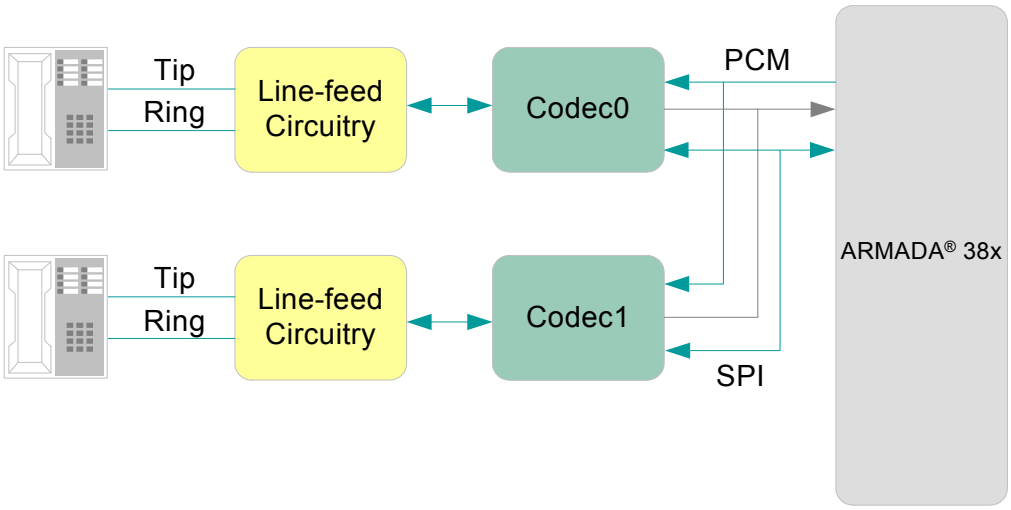
Copyright © 2015 Marvell

16 Time Division Multiplexing (TDM) Controller

This section describes the device Time Division Multiplexing (TDM) controller. The TDM controller is a 2-channel TDM controller (also referred to as TDM2C). This controller is used for interfacing with external Subscriber Line Interface Circuit (SLIC)/Subscriber Line Audio-processing Circuit (SLAC) /Direct Access Arrangement (DAA)/codec devices, as shown in Figure 48. This interface is useful for Voice over IP (VoIP) applications.

The TDM registers are located in Appendix A.6, Time Division Multiplexing Registers, on page 552.

Figure 48: SLIC/SLAC/DAA/Codec Connection Example



16.1 Features

The main features of the TDM controller are:

- Two voice channels (interfaces two SLIC devices)
- TDM interface with up to 128 full-duplex time slots; selectable time slot per each SLIC device
- Support for various bit clock rates (256 kHz to 8.192 MHz in increments by powers of two).
- TDM as master or slave of frame sync and PCM (Pulse-Code Modulation) clock
- Support for compound (A-law/U-law) or linear voice samples
- Support for wideband voice channels
- Support for various flavors of PCM (short and long frame synchronization, inverted frame synchronization, positive-edge/negative-edge PCM data drive/sample and others)

Copyright © 2015 Marvell

16.2 Functional Description

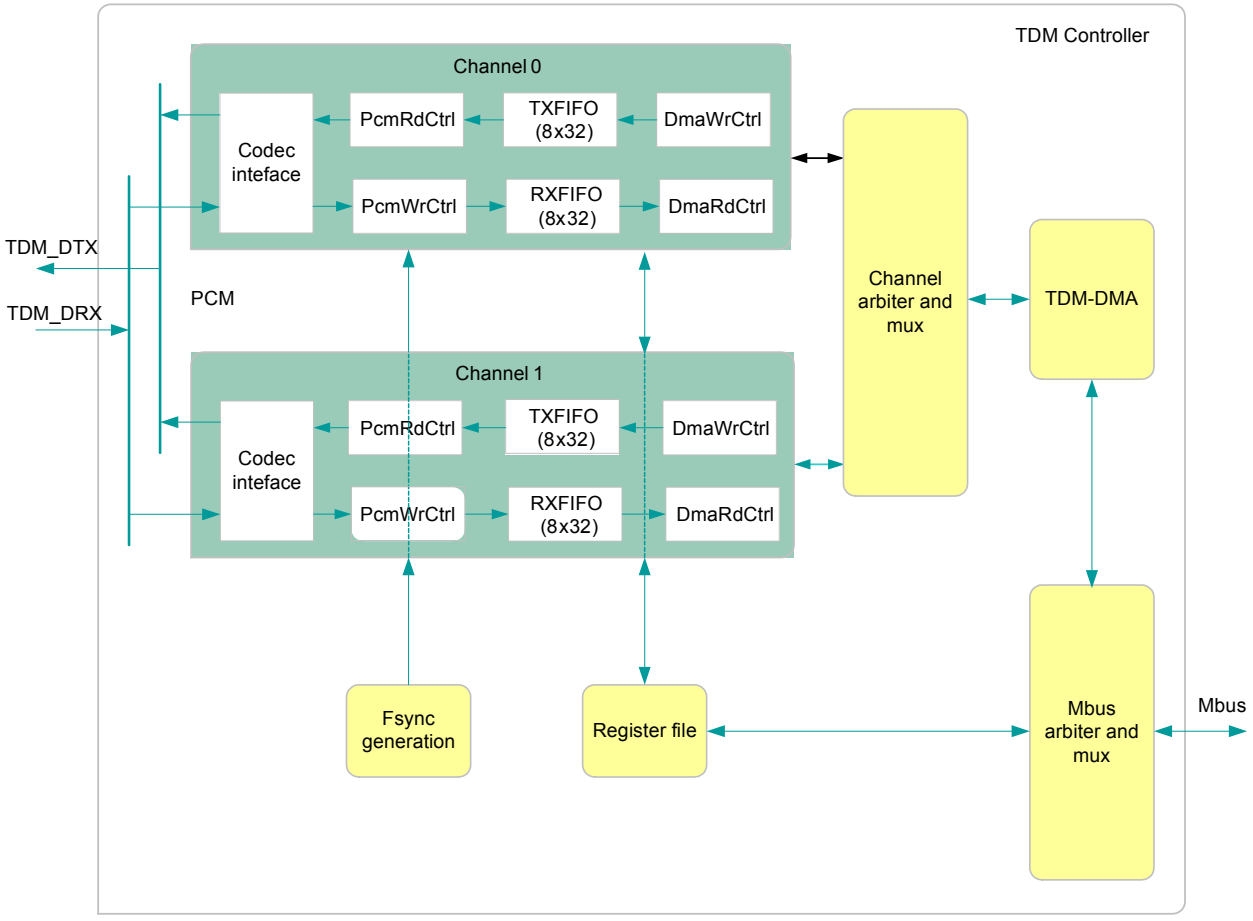
The TDM interface defines up to 128 full-duplex time slots (at 8.192 MHz PCM clock).
The TDM interface provides two independent transmit and two receive channels internally, via a tightly integrated TDM-DMA engine (see Figure 49). These channels can be mapped into one of the 128 (maximum) time slots contained in a frame, as defined in the control registers. CPU intervention is required only for overflow/underflow management and timely, buffer management.

The TDM interface supports the following functionality:

- PCM bus interface for telephony SLIC/SLAC/DAA/codec devices
- Internal register read/write access
- Interrupt generation to the CPU
- TDM-DMA operation to move data between the PCM bus and memory

Note Use the SPI controller for codec register read/write access (see Section 21, Serial Peripheral Interface (SPI)).

Figure 49: TDM Controller Block Diagram



Copyright © 2015 Marvell

The size and location of the buffers are programmable through the registers. If the TDM-DMA engine fails to service the data moving requirements of the TDM interface in a timely fashion (due to increased internal bus latencies), an underflow/overflow occurs, and the TDM interface generates a CPU interrupt and signals the error via the Interrupt Status Register (Table 342 p. 565) (ISR). The channel in which the error occurs is then switched off. In cases of Tx underflow, the TDM sends 0's on a programmed time slot. In cases of Rx overflow, it discards incoming data from the programmed time slot.

The TDM interface signals are provided in Table 44.

Table 44: Time Division Multiplexing (TDM) Interface Signals

Pin Name	I/O	Description
TDM2C_INTn	I	Interrupt Signal FROM the SLIC/codec
TDM2C_RSTn	O	SLIC/codec Reset Signal
TDM2C_PCLK	I/O	PCM Audio Bit Clock
TDM2C_FSYNC	I/O	TDM Frame Sync Signal
TDM2C_DRX	I	PCM Audio Input Data (for recording)
TDM2C_DTX	O	PCM Audio Output Data (for playback)



Note

These pins are multiplexed on the MPP pins. For further information, see the Pin Multiplexing section in the respective device *Hardware Specifications*.

The Frame Sync (FS) is generated (or sampled) indicating the start of the time slot. The time slot corresponds to one sample of 1 byte (A-law or U-law encoded) or 2 bytes (linear encoded). For 2 bytes, each sample occupies two adjacent time slots. In transmit operation, the data is driven out to the codec on the Data Tx (DTX) line. The TDM can drive data on the positive edge or negative edge of PCLK. The TDM can be the master of FS and PCLK (it drives both of these) or it can be the slave. As a slave, it receives PCLK and FS from an outside master.

16.3 TDM Protocol Specification

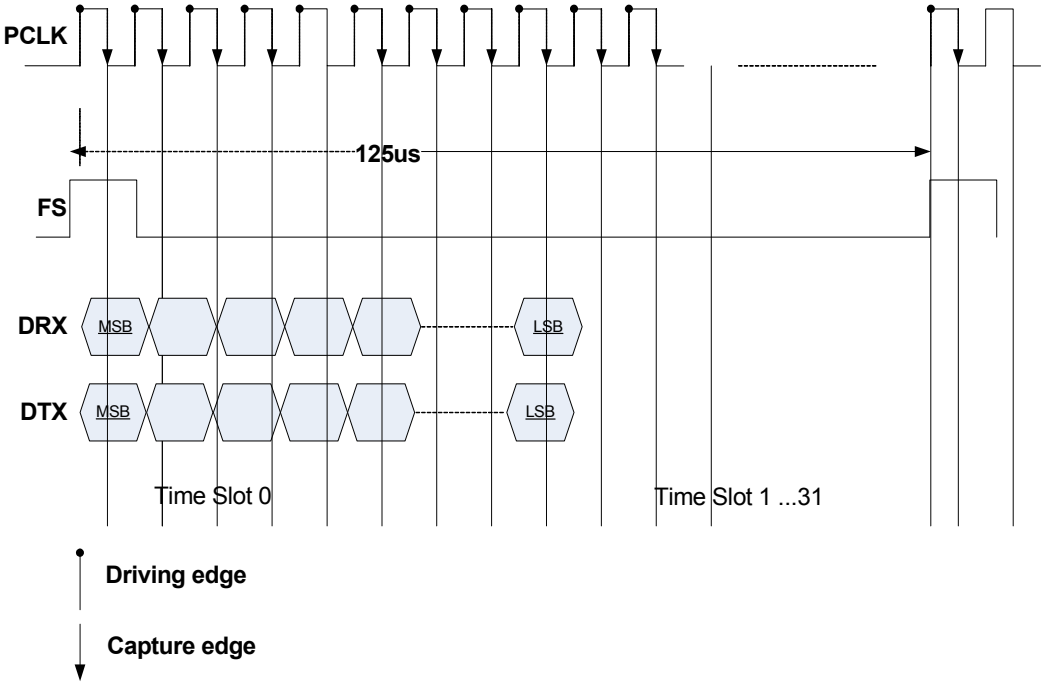
The TDM interface defines time slots for PCM data transmit and receive. The start of the time slot is indicated by an FS. The time slot consists of 8 PCM clocks (PCLK). The FS is always asserted at an interval of 125 µs, which corresponds to a frequency of 8 kHz. In each FS, one sample of PCM data is received and transmitted.

The FS indicates the start of the Time Slot 0, as shown in the Figure 50.

- **Data Rx –** The codec drives the Data Rx (DRX) line on the positive edge of the PCLK, and the TDM captures on the negative edge of the PCLK. In receive operations, the data is received from the codec on the DRX line serially, in MSB first order.
- **Data Tx –** The TDM drives data on the positive or negative edge of PCLK on the DTX line. The codec captures data on the negative edge of the PCLK. In transmit operations, the data is driven out to the codec on DTX serially, in MSB first order.

Copyright © 2015 Marvell

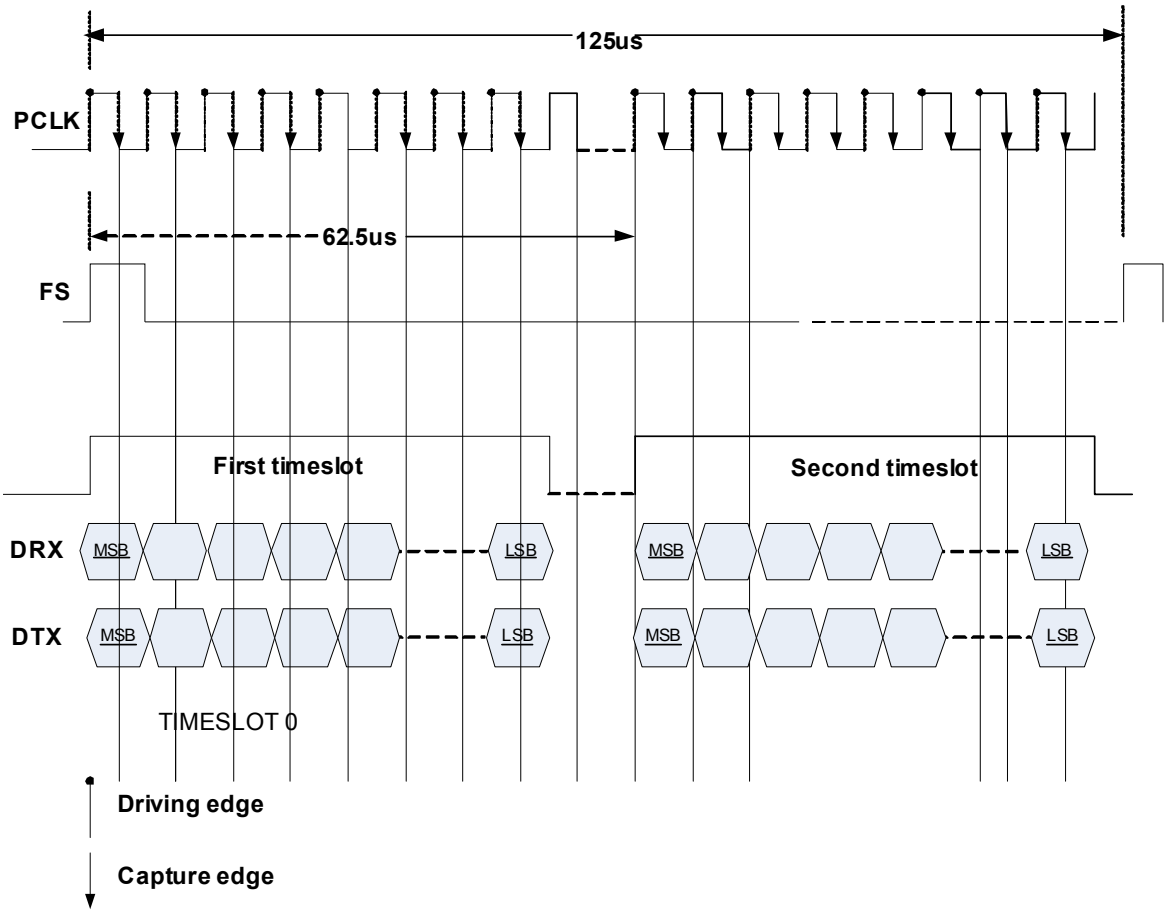
Figure 50: TDM Operation Time Slot 0



TDM also supports Wideband mode, in which two PCM samples are transmitted and received in one frame sync (125 µs) from TDM to codec and vice-versa. The wideband codec is used for enhanced voice quality in VoIP networks. The voice quality is improved because the effective sampling rate becomes 16 kHz (instead of the 8 kHz rate in the standard Narrowband mode). The second sample is sent in a time slot that is 62.5 µs from the first time slot, as shown in Figure 51.

Copyright © 2015 Marvell

Figure 51: TDM Wideband Mode Operation



The TDM drives data on the positive or negative edge of PCLK on the DTX line. The FS can be short, long, inverted, and driven on the positive or negative edge of PCLK, depending on the configuration.

16.3.1 Tx Data Flow

Note The term CHx in the following sections stands for CH0 or CH1 (Channel 0 or Channel 1).

The TDM supports two channels for Tx operation. Both channels are independent of one another, and each channel has a separate set of control and configuration registers.

The sequence for sending PCM data from the TDM to the codec is:

1. Firmware processes the incoming data from the IP network and creates a sample buffer in memory (typically a 10-ms sample buffer). The sample size must be in multiples of 4 bytes. The maximum size is the 30-ms sample period.
2. CPU initializes the TDM.

Copyright © 2015 Marvell

These registers are programmed only at the beginning of the Tx operation. Changing register values in the middle of TDM operation is prohibited.

- TDM PCM Clock Rate Divider Register (Table 338 p. 564)—selecting PCM clock.
- Number of Time Slots Register (Table 337 p. 563)—FS generation.
- Miscellaneous Control Register (Table 344 p. 566)—asserting reset to codec.
- PCM Control Register (Table 326 p. 557)—TDM features supported by TDM. By default, bits[1:0] are set to 0x3, which indicates target device mode TDM operation. For initiator device operation mode, set bits[1:0] to 0x0. The codec is then reset by programming the <CODEC_RST> field in the Miscellaneous Control Register (Table 344 p. 566) to allow sufficient time for the codec reset (see individual codec specification for details). The codec checks and loads the ratio of the PCM and FS clocks during reset, and uses the values for internal operation.
- Channel Time Slot Control Register (Table 327 p. 559)—selecting the time slot in which PCM data is received. The same time slot value should also be programmed in the codec. The Rx time slot of the TDM is the Tx time slot of the codec.
- Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563)—program with the <CHn_TOTAL_ SMPL_CNT> and <CHn_INT_SMPL_CNT> fields that are used by the TDM to synchronize with firmware. Firmware typically creates 10-ms sample width (the width may be any Dword size buffer up to the 30-ms sample width) of the buffer in memory. The sample from the codec can be 1- or 2-bytes in size, depending upon codec register settings.

3. CPU implements ping-pong buffers and enables transmit operation in TDM.
   These registers can also be programmed during TDM operation.

   - The CPU checks the <TX_DMA_ST_ADDR_OWN_CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) for a value 0. A value of 0 indicates that software can program the <TX_DMA_ST_ADDR_CH0> field in the Channel 0 Transmit Data Start Address Register (Table 332 p. 562) with the buffer address.
   - The CPU programs the <TX_DMA_ST_ADDR_OWN_CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) with 1, which indicates this buffer is now owned by the hardware.
   - The hardware makes a copy of the buffer address as programmed in the <TX_DMA_ST_ADDR_CH0> field in the Channel 0 Transmit Data Start Address Register (Table 332 p. 562), and then resets the <TX_DMA_ST_ADDR_OWN_CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) to 0.
     When this bit is not 0, the CPU cannot program the <TX_DMA_ST_ADDR_OWN_CHn> field with a new value. In this way, the ping-pong buffers can be implemented in memory by software.
   - The <TX_DMA_ST_ADDR_OWN_CHn> field points to the address of the Tx buffer in memory.
   - Codec register initialization: See the individual codec specification for register addresses and programming values.
   - The CPU programs the <CHnTxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to enable the channel transmit operation.

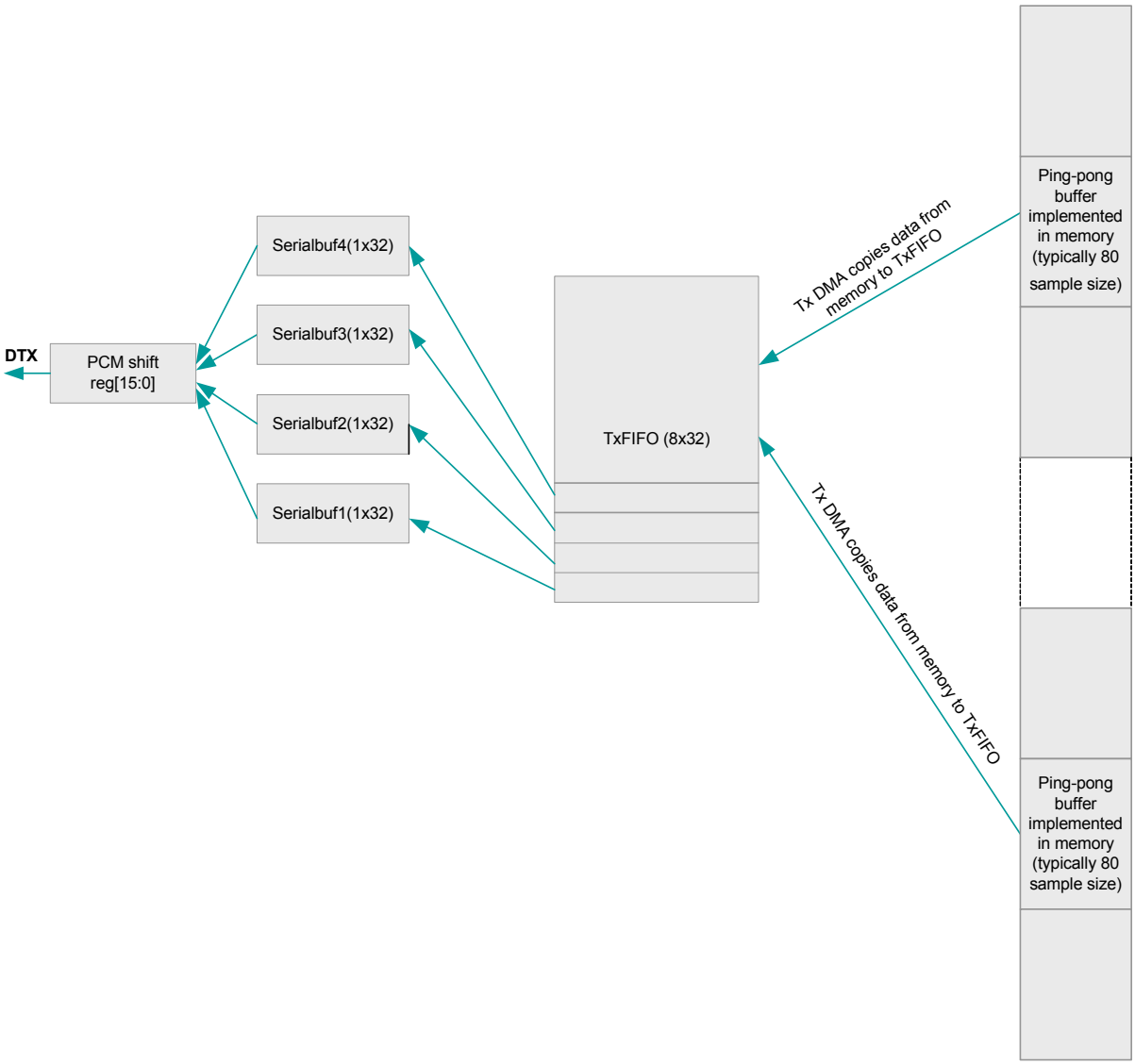
Note

The<TX_DMA_ST_ADDR_CH0> field must be 32-byte aligned.

Document Classification: Proprietary Information

4. TDM-DMA is accessed.
   - Active channels request access to TDM-DMA.
   - TDM-DMA performs a burst read of 32 bytes from memory.
   - Channel TXFIFO is filled up in one burst.
5. Once the TXFIFO is full, the TDM Tx read controller reads the TXFIFO and copies four Dwords to the serial buffer as shown in Figure 52. When the TXFIFO becomes empty, the TDM-DMA is activated again, to fetch the next 32 bytes from memory. This procedure continues until the total sample count is reached (see the <CHn_TOTAL_ SMPL_CNT> field in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563)). For example, if the <CHn_TOTAL_ SMPL_CNT> field is set to 80 bytes, the TDM-DMA is activated three times—fetching the first 32B, fetching the next 32B, fetching the last 16B.

Figure 52: TDM Transmit Path



6. As the sample sent to the codec becomes equal to the <CHn_INT_SMPL_CNT> field in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563), an interrupt—the <SCOCH0_TX_ INT> or <SCOCH1_TX_ INT> field in the Interrupt Status Register (Table 342 p. 566)—is issued to the CPU indicating the need for a new buffer. When the CPU receives the SCOCHx_TX_INT interrupt, three conditions arise:
   - Channel active: If the channel is still active, the CPU makes another buffer with the buffer size as programmed in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563). The CPU checks if the <TX_DMA_ST_ADDR_OWN_CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) is 0, programs the <TX_DMA_ST_ADDR_CH0> field in the Channel 0 Transmit Data Start Address Register (Table 332 p. 562) with the new buffer address, and sets the buffer ownership to 1.
     When all the data is fetched from the current buffer, the TDM checks that the <TX_DMA_ST_ADDR_OWN_CHn> field is 1 and makes a copy of the <TX_DMA_ST_ADDR_CH0> field for the next TDM-DMA operation. It also resets the <TX_DMA_ST_ADDR_OWN_CHn> field to 0 when the previous buffer operation is complete (i.e., all the data has been send out on the PCM bus). In this way, the software can poll the <TX_DMA_ST_ADDR_OWN_CHn> field to check the status of the buffer and can implement the ping-pong buffer.
     For normal operation, when the channel is active, Steps 4, 5, and 6 are repeated.
   - Channel closed on the VoIP side: Configure the <CHnTxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to close the channel. The TDM sends all the samples in the existing buffer to the PCM interface and then moves to IDLE state. The TDM also, generates the CHx_TX_IDLE interrupt—the <CH0_TX_IDLE> or <CH1_TX_IDLE> field in the Interrupt Status Register (Table 342 p. 565).
   - The CPU fails to create a new buffer (if the channel is active) or fails to close the channel: The TDM underflows (see Section 16.3.1.1, Tx Underflow, on page 200).

16.3.1.1 Tx Underflow

An underflow occurs when TDM is unable to meet the PCM transmit rate (one PCM sample every 125 µs in the Narrowband mode, or two samples in Wideband mode). There are two cases that might cause Tx underflow: TDM-DMA underflow or software underflow.

Hardware

During normal Tx operation, if the TXFIFO empties, because the last 4 Dwords are copied to the serial buffer, and there are still samples to be fetched from the buffer in memory, the TDM-DMA is triggered to fill the TXFIFO again. At that point, there are 17 PCM samples stored in the TDM to be sent on the DTX line (1 in the PCM shift register and 16 in the serial buffers). If the TDM-DMA fails to fetch the next 32 bytes from memory, before these 17 PCM samples are sent on the DTX line (17x125 µs), an underflow flag is set, which indicates that Tx underflow has occurred in the TDM. However, the underflow interrupt is not triggered until the TDM-DMA fills the TXFIFO. This delay occurs to avoid the TDM-DMA filling the TXFIFO at some later time and causing the TXFIFO write pointer to change.

See Actions to Perform for the steps necessary to implement recovery from hardware underflow.

Software

Software underflow occurs when the CPU fails to make a new buffer (if the channel is active) or fails to close the channel.

Once the SCOCHx_TX_INT is generated, there is a time window available for software to either point to a new buffer to TDM or to close the channel, if it is closed from the VoIP side. This window starts from the assertion of interrupt SCOCHx_TX_INT and ends at the buffer switch condition. The buffer switch condition is defined as "The existing buffer is already fetched to TXFIFO earlier, data from the TX FIFO is copied to serial buffers, and now there are four samples (500 µs in terms of

Document Classification: Proprietary Information

time) of PCM data remaining in the serial buffers." The TDM switches to the next buffer at this time and checks the <TX_DMA_ST_ADDR_OWN_CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) for a value of 1. The TDM needs to switch at this time to a new buffer so that it can perform the next TDM-DMA operation, in a timely manner, and fill the TX FIFO again. If at the buffer switch event, the TDM does not see <TX_DMA_ST_ADDR_OWN_CHn> = 1 and the channel is still enabled, the underflow flag is set.

Setting of this flag means that the TDM has underflow. However, the UFLOW_CHx_INT is still not generated, as the TDM has to send all the PCM samples that it is still storing. At this moment, there are 4 or 8 samples stored in the TDM (one in the PCM shift register and three in the serial buffer for Narrowband mode, 1 in the PCM shift register and 7 in the serial buffer for Wideband mode). After sending these samples out on the DTX line, the UFLOW_CHx_INT is triggered.

See Actions to Perform for the steps necessary to implement to recover from software underflow.

Actions to Perform

The following actions are performed by TDM in case of underflow:

1. The TDM generates an UFLOW_CHx_INT interrupt—the <UFLOW_CH0_ INT> or <UFLOW_CH1_ INT> field in the Interrupt Status Register (Table 342 p. 566)—to the CPU if the interrupt is enabled in the Interrupt Status Register (Table 342 p. 565) and Interrupt Status Mask Register (Table 340 p. 564).
2. The TDM switches off the channel transmit by setting the <CHnTxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to 0 and moves to IDLE state. It also flushes the TXFIFO. It keeps sending 0s on the PCM interface. The audio is disrupted.

To start the transmit operation again, software has to re-initialize the channel, by following Step 3 in Section 16.3.1, Tx Data Flow, on page 197.

16.3.1.2 Setting INT_SAMPLE_CNT

It is important to set the <CHn_INT_SMPL_CNT> field in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563) for normal Tx operation of the TDM. Once this number of samples is sent on the DTX line, the SCOCHx_TX_INT interrupt is generated. Software then has a window of time to point to the new buffer or close the channel, if it is closed from the IP side. Software should set the <CHn_INT_SMPL_CNT> field to allow the CPU sufficient time to point to the new buffer or close the channel. This ensures that a Tx underflow does not occur.

In quantitative terms the window size is given by the following equation (see the software underflow description in Section 16.3.1.1, Tx Underflow, on page 200):

- For Narrowband mode(1-byte or 2-byte sample)
  Window size = (TOTAL_SAMPLE_CNT - 4 - INT_SAMPLE_CNT) * 125 µs
- For Wideband mode (1-byte or 2-byte sample)
  Window size = (TOTAL_SAMPLE_CNT - 8 - INT_SAMPLE_CNT) * 125 µs

16.3.2 Rx Data Flow

The TDM supports two channels for Rx operation. Both channels are independent of one another, each with a separate set of control and configuration registers.

The sequence for receiving PCM data from the codec (codec to TDM) is:

1. When the telephone is picked up to start a conversation, an off-hook state is detected, and the codec generates an interrupt. The dial tone is passed to the POTS by the codec, and the CPU reads the interrupt. Once the receive channel is active:
   - A codec interrupt is indicated by the <CODEC_INT> field in the Interrupt Status Register (Table 342 p. 565).
   - The CPU reads the Interrupt Status Register (Table 342 p. 565) in the codec.

Copyright © 2015 Marvell

- The receive channel is now active.
- The CPU starts the TDM initialization process.

2. CPU initializes the TDM.

   These registers are programmed only at the beginning of the Rx operation. Changing register values in the middle of TDM operation is not permitted.

   - TDM PCM Clock Rate Divider Register (Table 338 p. 564)—selecting PCM clock
   - Number of Time Slots Register (Table 337 p. 563)—FS generation
   - Miscellaneous Control Register (Table 344 p. 566)—assessing reset to codec
   - PCM Control Register (Table 326 p. 557)—TDM features supported by TDM.
     By default, bits[1:0] are set to 0x3, which indicates target device mode TDM operation. For initiator device operation mode, set bits[1:0] to 0x0. The codec is then reset by programming the <CODEC_RST> field in the Miscellaneous Control Register (Table 344 p. 566) to allow sufficient time for the codec reset (see individual codec specification for details). The codec checks and loads the ratio of PCM and FS clocks during reset and uses the values for internal operation.
   - Channel Time Slot Control Register (Table 327 p. 559)—selecting the time slot in which PCM data is received. The same time slot value should also be programmed in the codec. The Rx time slot of the TDM is the Tx time slot of the codec.
   - Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563)—program with the <CHn_TOTAL_ SMPL_CNT> and <CHn_INT_SMPL_CNT> fields, which are used by the TDM to synchronize with the firmware. Firmware typically makes 10-ms sample width (The width may be any Dword size buffer up to the 30-ms sample width.The size must be a multiple of 4 bytes.) of buffer in memory. The sample from the codec can be 1- or 2-bytes in size, depending upon codec register settings.

3. CPU implements ping-pong buffers and enables receive operation in the TDM.

   These registers can also be programmed during TDM operation.

   - The CPU checks the <RX_DMA_ST_ADDR_OWN_ CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) for a value 0. A value of 0 indicates that software can program the <RX_DMA_ST_ADDR_CH0> field in the Channel 0 Receive Data Start Address Register (Table 333 p. 562) with the buffer address.
   - The CPU programs the <RX_DMA_ST_ADDR_OWN_ CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) with 1, which indicates that this buffer is now owned by hardware.
   - TDM makes a copy of the buffer address as programmed in the <RX_DMA_ST_ADDR_CH0> field, and then resets the <RX_DMA_ST_ADDR_OWN_ CHn> field to 0. When this bit is not 0, the CPU cannot program the <RX_DMA_ST_ADDR_CH0> field with a new value. In this way, ping-pong buffers can be implemented in memory by software.
   - <RX_DMA_ST_ADDR_CH0>—points to the address of the Rx buffer in memory.
   - Codec register initialization: See the individual codec specifications for register addresses and programming values.
   - The CPU programs the <CHnRxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to enable the channel receive operation.

   NOTE: The <RX_DMA_ST_ADDR_CH0> field must be 32-byte aligned.

4. Once the channel is enabled, the TDM begins receiving data serially (from the next FS) on the DRX line in the programmed time slot as shown in Figure 53. Once a voice sample is received in PCM shift register, it is copied to a serial buffer. This serial buffer helps to overcome the latency on the internal bus (backup buffer). There are four serial buffers (1 Dword in size each). Filling up one or all of these buffers is set by the <PerfBit> field in the PCM Control Register

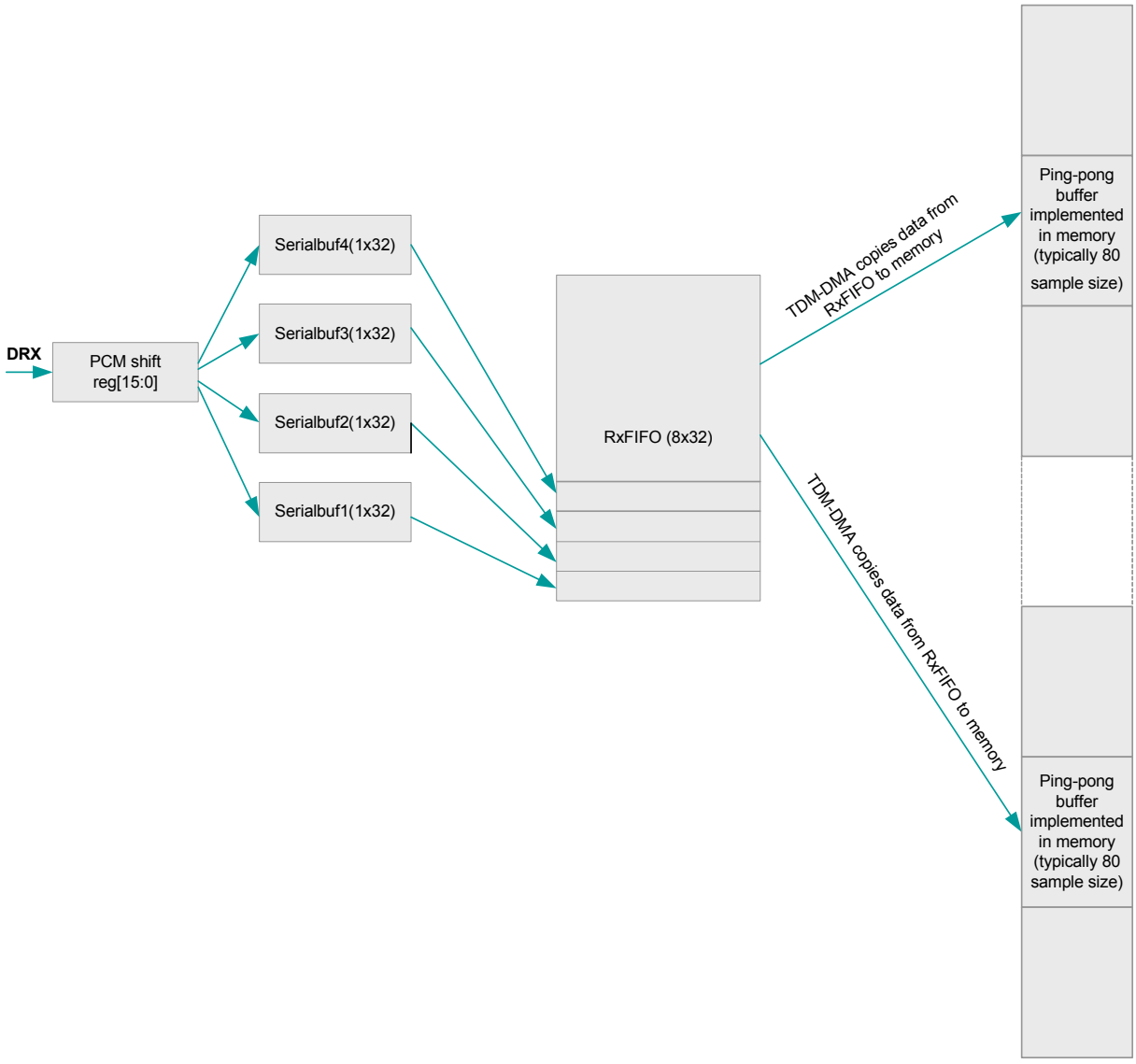
Copyright © 2015 Marvell
Document Classification: Proprietary Information

(Table 326 p. 557). If the <PerfBit> field is set to 1 (default), all serial buffers are filled with PCM samples, and then they are copied to the RXFIFO.

5. Once the RXFIFO becomes full, the TDM-DMA is accessed:
   - The channel requests access to the TDM-DMA.
   - The TDM-DMA performs a burst write of 32 bytes to memory (the channel RXFIFO is emptied in one burst).

   When the RXFIFO becomes full, the TDM-DMA is activated again to write the next 32 bytes to memory. This procedure continues until the total sample count is reached. For example, if the <CHn_TOTAL_ SMPL_CNT> field in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563) is set to 80 bytes, the TDM-DMA will be activated three times—write first 32 bytes, write next 32 bytes, write last 16 bytes.

Figure 53: TDM Receive Path



6. As samples are received from codec, and the number becomes equal the <CHn_INT_SMPL_CNT> field in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563) value, a SCOCHx_RX_INT interrupt is issued to the CPU indicating that a new buffer should be created. The CPU evaluates the conditions that can arise during TDM Rx operations.
   - Channel is still active—If the channel is still active, the CPU creates another buffer of the same size as that programmed in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563) (after receiving a SCOCHx_RX_INT interrupt). The CPU checks the <RX_DMA_ST_ADDR_OWN_ CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) for a value of 0, programs the <RX_DMA_ST_ADDR_CH0> field in the Channel 0 Receive Data Start Address Register (Table 333 p. 562) with the new buffer address, and sets the buffer ownership to 1.
     When all the PCM data is written to the current buffer, the TDM then checks that the <RX_DMA_ST_ADDR_OWN_ CHn> field is 1 and makes a copy of the <RX_DMA_ST_ADDR_CH0> field for the next TDM-DMA operation, and reset the <RX_DMA_ST_ADDR_OWN_ CHn> field to 0. In this way, software can poll the <RX_DMA_ST_ADDR_OWN_ CHn> field to check the status of the buffer and to start further processing of the buffer.
     For normal operation, when the channel is active, Steps 4, 5, and 6 are repeated.
   - Channel closed due to telephone hang up (on-hook)—The user can hang up the telephone at any time during normal Rx operation. This can occur before the buffer is completely full. When the telephone hangs up, the codec generates an interrupt to the CPU (the <CODEC_INT> field in the Interrupt Status Register (Table 342 p. 565)). After receiving the interrupt, the CPU can close the channel Rx by programming the <CHnRxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to 0. In this case, the TDM stops receiving data from the DRX line, and the remaining PCM samples in the buffer are filled with dummy data (programmed in the <DUMMY_DATA> field in the Dummy Data for Dummy RX Write Register (Table 343 p. 566)). The TDM then enters an IDLE state and generates a CHx_RX_IDLE interrupt.
   - The CPU fails to make a new buffer (if the channel is active) or fails to close the channel after the telephone is hung up—the TDM overflows (see Section 16.3.2.1, Rx Overflow, on page 204).

16.3.2.1 Rx Overflow

If the TDM is unable to meet the PCM transmit rate (one PCM sample every 125 µs in Narrowband mode, or two samples in Wideband mode), an overflow occurs. There are two cases that might cause Rx overflow—TDM-DMA overflow or software overflow.

Hardware

During normal Receive operation, the RXFIFO becomes full when the last 4 Dwords are copied from the serial buffers. The TDM-DMA is then triggered to empty the RXFIFO. From that moment, the TDM can store 17 PCM samples (9 samples for linear mode) coming from DRX line. It can store 1 PCM sample in the PCM shift register and 16 in the serial buffers, if using 1-byte samples (or 1 in the PCM shift register and 8 in the serial buffer, if using 2-byte samples). If the TDM-DMA fails to write the 32 bytes to memory before these 17 PCM samples are received on the DRX line (17 x 125 µs), an overflow flag is set, which indicates that RX overflow has occurred in the TDM. However, the overflow interrupt is not triggered until the TDM-DMA empties the RXFIFO. This is to avoid TDM-DMA reading the RXFIFO at some later time and causing the RXFIFO read pointer to change.

Document Classification: Proprietary Information

In case of overflow, the following actions are performed by the TDM:

1. The TDM generates an interrupt OFLOW_CHx_INT to the CPU if the interrupt is enabled in the Interrupt Status Register (Table 342 p. 565) and Interrupt Status Mask Register (Table 340 p. 564).
2. The TDM switches off the channel receive by setting the <CHnRxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to 0 and goes to IDLE state. The TDM also flush the RXFIFO. It stops receiving data from PCM interface. The buffer that was being processed at the time overflow occurred becomes invalid. The audio is disrupted.

To start the receive operation again, software has to re-initialize the channel by following the Step 3 in Section 16.3.2, Rx Data Flow, on page 201.

See Actions to Perform for the steps necessary to implement to recover from hardware overflow.

Software

Software overflow occurs when the CPU fails to create a new buffer (if the channel is active) or fails to close the channel after the telephone is hang up.

Once SCOCHx_TX_INT is generated, there is a time window available for software to either point to new buffer to TDM or to close the channel if it is closed from the IP side. This window starts from assertion of interrupt SCOCHx_RX_INT and ends at the buffer full condition. The buffer full condition is defined as "All the PCM samples of the existing buffer have been written to the buffer in memory." The TDM switches to the next buffer at that time and checks the <RX_DMA_ST_ADDR_OWN_ CHn> field in the Channel n Buffer Ownership Register (n=0–1) (Table 331 p. 561) for a value of 1 (to fetch the address of a new buffer). If the TDM detects <RX_DMA_ST_ADDR_OWN_ CHn> as 0 and the channel is still enabled, overflow interrupt OFLOW_CHx_INT is set.

See Actions to Perform for the steps necessary to implement to recover from hardware overflow.

Actions to Perform

The following actions are performed by the TDM in the case of overflow:

1. The TDM generates an interrupt OFLOW_CHx_INT to CPU if the interrupt is enabled in the Interrupt Status Register (Table 342 p. 565) and Interrupt Status Mask Register (Table 340 p. 564).
2. The TDM switches off the channel receive by setting the <CHnRxEn> field in the Channel n Enable and Disable Register (n=0–1) (Table 330 p. 561) to 0 and moves to IDLE state. It also flushes the RXFIFO. It stops receiving data from the PCM interface. The audio is disrupted.

To start the receive operation again, software has to re-initialize the channel by following the Step 3 in Section 16.3.2, Rx Data Flow, on page 201.

16.3.2.2 Setting Interrupt Sample Count

Setting the <CHn_INT_SMPL_CNT> field in the Channel n Total Sample Count Register (n=0–1) (Table 336 p. 563) plays an important role in normal Rx operation of the TDM. Once the number of samples set in that field are received on the DRX line, the SCOCHx_RX_INT interrupt is generated. Software then has a window of time to point to a new buffer or close the channel, if the user hangs up the telephone. Software should set <CHn_INT_SMPL_CNT> so that the CPU has sufficient time to point to the new buffer or close the channel. This ensures that a Rx overflow does not occur.

In quantitative terms, the window size is given by the following equation (see the software overflow description in Section 16.3.2.1, Rx Overflow):

- For 1 byte sample
  Window size = (TOTAL_SAMPLE_CNT - INT_SAMPLE_CNT) * 125 µs
- For 2 byte sample
  Window size = (TOTAL_SAMPLE_CNT - INT_SAMPLE_CNT) * 125 µs

Copyright © 2015 Marvell

16.3.3 TDM Wideband Mode Operation

In Wideband mode, two PCM samples are transmitted and received in one frame sync (125 µs) from TDM to codec and vice-versa. The wideband codec is used for enhanced voice quality in Voice over IP (VoIP) networks. The voice quality is improved because the effective sampling rate becomes 16 kHz. The second sample is sent in a time slot that is 62.5 µs from the first time slot.

Each of the two channels of the TDM can be individually configured in Wideband or Narrowband mode. The <CH0WBand> field and the <CH1WBand> field in the PCM Control Register (Table 326 p. 557) define the Wideband versus Narrowband mode selection. For the Wideband time slot programming, two registers are provided in each channel. The transmit and receive flow remains the same as explained in Section 16.3.1, Tx Data Flow, on page 197 and Section 16.3.2, Rx Data Flow, on page 201.

To set Channel x to Wideband mode:

1. Set the <CH0WBand> field in the PCM Control Register (Table 326 p. 557) to 1 to select Wideband mode for CHx.
2. Set the <CH0DlyEn> field in the PCM Control Register (Table 326 p. 558) to 1 to select delay control for time slot programming.
3. Program the Channel 0 Delay Control Register (Table 328 p. 560) for the first time slot for transmit and receive of PCM data.
4. Program the TDM Channel 0 Wideband Delay Control Register (Table 352 p. 569) for the second time slot for transmit and receive of PCM data. For wideband codecs, this time slot is generally 62.5 µs from the first time slot.

Document Classification: Proprietary Information

17 PCI Express Interface (PCIe) 2.0

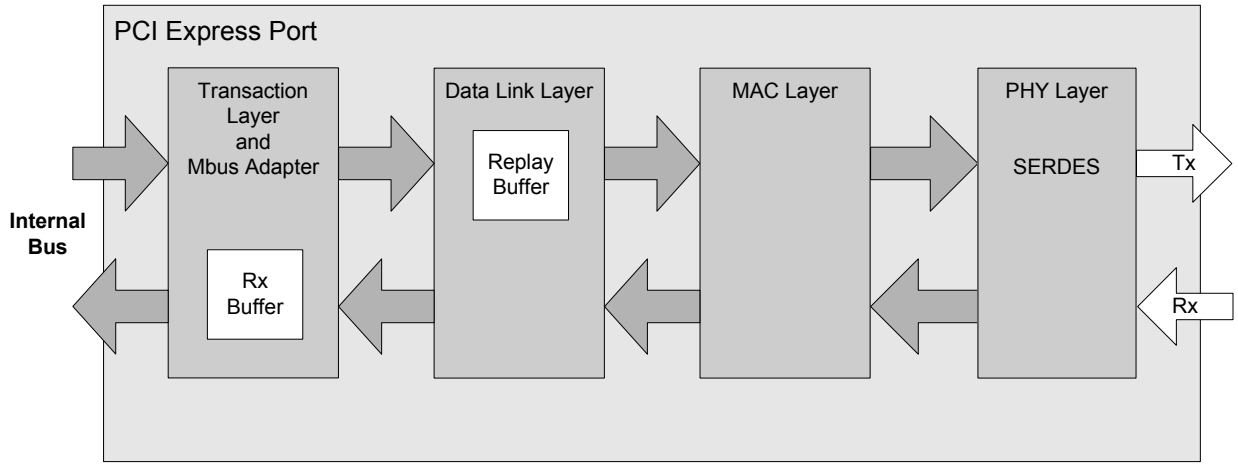
Figure 54 shows the connectivity of PCIe Express controller clusters to the Mbus fabric, lane multiplexing, and high-speed SERDES.

The PCIe Express cluster supports:

- **PCIe port 0:** A single x4 lanes PCIe Controller Gen2
  - Backward compatible as x1 lane mode
  - Root Complex (RC) or Endpoint mode (EP) mode
- **PCIe ports 1–3:** 3 additional x1 Lanes PCIe Controllers Gen2 RC only mode
- Multiplexed on High Speed SERDES

Figure 54 is a high-level diagram of a single PCI Express port.

Figure 54: High-level Block Diagram



The PCI Express 2.0 registers are located in Appendix A.7, PCI Express 2.0 Port 0 Registers.

17.1 Features

The PCI Express interface includes the following features:

- PCI Express Base 2.0 compatible
- Root Complex port
- Can be configured also as an Endpoint port
- Embedded PCI Express PHY based on proven Marvell® SERDES technology
- x4 link width, or x1 links when configured to quad mode
- Up to 5.0 GHz signaling
- Lane polarity inversion support
- Lane reversal support
- Replay buffer
- Maximum payload size of 128 bytes

Copyright © 2015 Marvell

- Single Virtual Channel (VC-0)
- Ingress and egress Flow Control
- Extended Tag support
- Interrupt emulation message support
- Power management support:
  - Software power management states D1, D2, D3 hot, and D3 cold
  - Active state power management L0s and L1.
- CLKREQn support as Root Complex
- Advanced Error Reporting (AER) capability support
- Single function device configuration header
- Message Signaled Interrupts (MSI) and MSIx support (MSIx support in Root Complex only)
- Support for Dynamic Link Width change
- Support for Dynamic Speed change
- Power Management (PM) capability support, as an Endpoint
- Programmable address map

Note

The <PCIe0 Intr Mask> field in the SoC Control 1 Register (Table 1888 p. 1395) should be cleared when PCIe Port 0 is configured to work as Endpoint.

17.2 Functional Description

The PCI Express interface uses a layered architecture, according to the PCI Express specifications. The main layers are the PHY layer, MAC layer, Data Link layer, and Transaction layer. In addition, a Core Adapter layer handles the forwarding of the PCI Express Transaction Layer Packets (TLP) to the device Mbus.

17.2.1 PHY Layer

On the Tx path, the PHY layer receives symbols from the MAC layer, converts them into a serialized format, and transmits them on the PCIe port. On the Rx path, the PHY layer receives a serialized stream from the PCI-E port, and forwards them as parallel symbols to the MAC layer.

The PHY handles symbol-locking and 8b/10b encoding/decoding.

The PHY layer is responsible also for the clock tolerance compensation. The received symbol stream is adapted to the local clock, by adding or deleting skip OSs (Ordered Sets).

17.2.2 MAC Layer

The MAC layer is responsible for compensating for the skew between the different lanes. This is called lane-to-lane deskew. A skew between different lanes can be caused by factors such as board routing.

The MAC layer is also responsible for establishing and maintaining the PCI Express link, packet framing, and data packing according to the link width. The PCI Express Link Training and Status State Machine (LTSSM) is located in this layer. The LTSSM contains the functionality for the link configuration in terms of the link width, lane polarity, and lane numbering.

Additionally, the MAC layer performs the scrambling functions. The MAC layer controls the different link power-management modes, loopback mode, link disable mode, and link hot-reset function. In addition, the MAC layer handles the generation and detection of the various TSs (Training Sequences) and OSs.

Document Classification: Proprietary Information

On the Tx path, the MAC layer receives packets (TLPs and DLLPs—Data Link Layer Packets) from the Data Link layer. The packets are framed, scrambled, and packed into the relevant link width and forwarded to the PHY.

On the Rx path, the MAC layer receives aligned symbols from the PHY. The symbols are un-packed according to the relevant link width and un-framing is performed. The packets (DLLPs and TLPs) are then extracted from the frames and forwarded to the Data Link layer.

17.2.3 Data Link Layer

The Data Link layer provides a reliable TLP exchange between two components on the PCIe link. This layer performs most of the data integrity functions as specified by the PCI Express 2.0 specification.

The Data Link layer controls the sequence number generation and detection. It also controls the LCRC generation and detection. Outgoing TLPs are temporarily stored in a replay buffer until an acknowledge is received from the far-end component. When a corrupted or missing TLP is detected, the replay mechanism is used to recover and maintain reliable Transaction layer to Transaction layer connection. The replay buffer holds transmitted packets and re-transmits them when required.

The Data Link layer handles the generation and processing of DLLPs. DLLPs are used for conveying information such as flow control, TLP acknowledgment, and power management handshake.

17.2.4 Transaction Layer

The Transaction layer primary responsibility is handling of TLPs. Outgoing TLPs are assembled and scheduled for transmission. Incoming TLPs are parsed and checked for various errors. In addition, the Transaction layer is responsible for handling the split transaction protocol—both towards the PCIe port and the internal bus.

The Tx path accepts TLPs from the Mbus and schedules them for transmission, based on the flow-control credit availability and the relevant ordering rules. Non-Posted (NP) TLPs are assigned with a unique tag before they are scheduled for transmission. TLPs are then passed on to the Data Link layer for transmission.

The Rx path examines the incoming TLPs for a variety of packet formation errors. Incoming completions tags are checked for a valid NP request, which was sent by the device. TLPs are then passed to the Mbus.

PCIe x4

On the Tx, the PCIe unit x4 ports support:

- 12 Posted or Completion TLPs, dynamically allocated
- 11 Non-Posted TLPs

On the Rx, the PCIe unit x4 ports support:

- 11 Completion TLPs
- 5 Posted TLPs
- 11 Non-Posted TLPs

PCIe x1

On the Tx, the PCIe unit x1 ports support:

- 6 Posted or Completion TLPs, dynamically allocated
- 8 Non-Posted TLPs

On the Rx, the PCIe unit x1 ports support:

- 4 Completion TLPs

Copyright © 2015 Marvell

- 4 Posted TLPs
- 8 Non-Posted TLPs

17.3 Link Initialization

Enable the PCI Express interface by setting the <PCIe0En>, <PCIe1En>, <PCIe2En>, or <PCIe3En> field in the SoC Control 1 Register (Table 1888 p. 1395). This allows programming of link parameters before the start of link initialization. The highest common link width is established according to the following order: x4 to x1.

Lane polarity inversion is supported. Each lane differential couple can be routed on the board regardless of its polarity. In addition, lane reversal is supported.

In case the initialization fails and no link is established, the PHY will keep on trying to initiate a link forever unless the port is disabled. As long as the port is enabled, the PHY will continue trying to establish a link; once the PHY identifies that a device is connected to it, a link will be established.

17.4 Disabling a Link

To disable a link, follow these steps:

- When Port0 is operating as x4:
  1. Set the <LnkDis> field in the PCI Express Link Control Status Register (Table 411 p. 602).
  2. To ensure that the link is disabled, poll the <DLDown> field in the PCI Express Status Register (Table 430 p. 616).
  3. Clear the <PCIe0En> field in the SoC Control 1 Register (Table 1888 p. 1396).
  4. Clear <LnkDis>.

- When Port0 is x1 or working with Ports 1-3:
  1. Set the <LnkDis> field in the PCI Express Link Control Status Register (Table 411 p. 602).
  2. Clear the <PCIeXEn> field in the SoC Control 1 Register.
  3. Clear the <LnkDis>.
  4. To ensure that the link is disabled, poll the <DLDown> field in the PCI Express Status Register (Table 430 p. 616).

To restore the connection, set the <PCIeXEn> field in the SoC Control 1 Register.

17.5 Master Memory Transactions

Master memory transactions are memory space read and write requests (MRd and MWr TLPs) that are generated and sent over the PCI Express link, and the respective completion TLPs that are received in return.

The following features are supported as a master memory requester:

- Port x.0: Eleven outstanding Non-posted (NP) requests (memory read requests) and up to eleven Posted (P) requests (memory write requests)
- Ports x.1/x.2/x.3: Four outstanding NP requests (memory read request) and up to five P requests (memory write request)
- Maximum memory read request of 128 bytes
- Maximum memory write request of 128 bytes
- 64-bit addressing

Copyright © 2015 Marvell

17.6 Master I/O Transactions

Master I/O transactions are I/O space read and write requests (IORd and IOWr TLPs) that are generated and sent over the PCI Express link, and the respective completion TLPs that are received in return.

The following features are supported as a master I/O requester:

- Port x.0: Eleven outstanding NP requests (I/O read or write)
- Ports x.1/x.2/x.3: Four outstanding NP request (I/O read or write)
- Maximum I/O read request of 4 bytes
- Maximum I/O write request of 4 bytes
- 32-bit addressing



Note

- Do not initiate I/O requests that are larger than 4 bytes and cross the 4 byte address boundary. These requests are illegal according to the PCI Express Base 2.0 specifications.
- In Endpoint mode, only memory request generation is allowed by the PCI Express Base 2.0 Specification. When working in Endpoint mode, do not initiate IO requests.

17.7 Master Configuration Transactions

Master Configuration transactions are configuration space read and write requests (CfgRd0, CfgWr0, CfgRd1 and CfgWr1 TLPs) that are generated and sent over the PCI Express link, and the respective completion TLPs that are received in return.

The following features are supported as a master configuration requester:

- Port x.0: Eleven outstanding NP requests (configuration read or write)
- Ports x.1/x.2/x.3: Four outstanding NP request (configuration read or write)
- Maximum Configuration read request of 4 bytes
- Maximum Configuration write request of 4 bytes
- Extended register number support (4 KB extended PCI Express configuration header space)



Note

In Endpoint mode, only memory request generation is allowed by the PCI Express Base 2.0 Specification. When working in Endpoint mode, do not initiate configuration requests.

Generation of Configuration Requests

As a Root Complex port, the core processor may generate Type0 or Type1 configuration cycles to the PCI Express Endpoints, via indirect access using the PCI Express Configuration Address Register (Table 385 p. 585) and PCI Express Configuration Data Register (Table 386 p. 585). The following procedure is used for generating configuration cycles:

1. PCI Express Configuration Address Register—Write the Target Bus, Device, Function, Register and Extended Register Numbers fields, using the <ConfigEn> field to enable this mechanism.
2. PCI Express Configuration Data Register—Read or write to generate a respective read or write configuration request. The type of the request (type 0 or type 1) is set according to the following rules:
   - *Type1 request:* generated if Target Bus Number Is different from the internal Bus Number.
   - *Type0 request:* generated if Target Bus Number is same as the internal Bus Number, and the Target Device Number is different from the internal Device Number.

Copyright © 2015 Marvell

The transmitted Configuration TLP includes the Target Bus, Device, Function and Register Numbers as written to the PCI Express Configuration Address Register.

The Configuration request generation is only enabled when the <ConfigEn> bit is set.

17.8 Target Memory Transactions

Target Memory transactions are memory space read and write requests (MRd, MWr TLPs) that are received over the PCI Express link, and the respective completion TLPs that are generated and transmitted in return.

The following features are supported as a target memory completer:

- Port x.0: Reception of up to eleven Memory read requests
- Ports x.1, x.2 and x.3: Reception of up to eight Memory read requests
- Port x.0: Reception of up to five Memory write transactions
- Ports x.1, x.2, and x.3: Reception of up to four Memory write transactions
- Maximum received read request size of 4 KB
- Maximum received write request of 128 bytes
- Support PCI Express access to all of the device's internal registers
- 64-bit addressing
- Three Memory BARs (64-bit), BAR0 is dedicated to internal register access
- In Endpoint mode: Expansion ROM support

17.9 Target I/O Transactions

Target I/O transactions are I/O space read and write requests (IORd, IOWr TLPs) that are received over the PCI Express link, and the respective completion TLPs that are generated and transmitted in return.

Target I/O transactions are not supported by the device and should not be generated by the downstream device.

17.10 Target Configuration Transactions

Target Configuration transactions are Configuration space read and write requests (CfgRd0, CfgWr0, CfgRd1 and CfgWr1 TLPs) that are received over the PCI Express link, and the respective completion TLPs that are generated and transmitted in return.

When configured as Root Complex port, target Configuration transactions are not supported and should not be generated by downstream device. In Endpoint mode, Target Type0 Configuration transactions are supported.

The following features are supported as a target Configuration completer:

- Reception of up to eleven NP requests (configuration read or write) for ports x.0 and up to eight for the rest.
- Maximum received configuration read request size of 4 bytes.
- Maximum received configuration write request of 4 bytes.

The device as an end point supports responding with Configuration Request Retry Status (CRS) to a configuration read/write access. This is useful for postponing the root complex access till the local CPU finish initialization of the end point.

To enable this feature, set the <CRS Enable> field in the PCI Express Control Register (Table 429 p. 613) to 1 prior to enabling link training (meaning, prior to setting the <PCIe0En> field in the SoC Control 1 Register (Table 1888 p. 1396). When the local CPU finishes end point initialization, clear the <CRS Enable> bit.

Document Classification: Proprietary Information

17.11 Target Special Cases

- Access attempts that fail address decoding (for example, do no hit a memory BAR) are completed as Unsupported Requests.
- MemWr accesses to reserved, or not implemented registers, are completed normally on the PCI Express port, and the data is discarded.
- MemRd accesses to reserved, or not implemented registers, are completed normally on the PCI Express port, and a CplD TLP with data value of 0 and *SC* (*Successful Completion* status) is returned.

17.12 Messages

PCI Express defines a new message space. Messages are used to replace legacy PCI side-band signals such as interrupts, error signals, hot-plug signals etc. Messages are also used to enable new capabilities such as active power management, Slot Power Limit and others.

Table 45 lists the message groups supported as a Root Complex port, and if the group is supported by the device.

Table 45: Supported Message Groups—Root Complex Mode

Message Group	Supported	Action
Interrupt Signaling	Yes	A received *Assert_INTx* message is forwarded as an interrupt to the core processor. Reception of a *Deassert_INTx* message clears the relevant interrupt. **NOTE:** Both INTA, INTB, INTC and INTD are supported. (x = A, B, C or D)
Power Management Event (PME)	Yes	A received PM_PME message is forwarded as an interrupt to the core processor. The log of the message is registered in PCI Express Root Complex Power Management Event (PME) Register. While software is handling one PME message, a new PME message may be received and registered in this register.
Error Signaling	Yes	A received Error message is forwarded as an interrupt to the core processor. Both Correctable, Non-fatal and Fatal error messages are supported.
Hot Plug Signaling	No	
Locked Transaction Support	No	
Slot Power Limit Support	Yes	To enable sending SSPL, set the <SsplMsgEnable> field in the PCI Express Root Complex Set Slot Power Limit (SSPL) Register (Table 432 p. 616) bit[16] to 1 before link is up. If enabled, upon reaching Link_up and whenever there is a change in the <SlotPowerLimitValue> bits[7:0] or <SlotPowerLimitScale> bits[9:8] an SSPL message is transmitted.
PME_TurnOff	Yes	To send a TurnOff message, set the <Send Turn Off Msg> field in the PCI Express Power Management Register (Table 435 p. 618) bit[5] to 1. When the EP reaches L2 state, the <EPReady4TurnOff> interrupt is asserted, indicating to the host that it may turn off the primary power of the EP. To exit this state, the software must initiate soft reset and clear the <SndTurnOff> bit once the link is up again.

Copyright © 2015 Marvell

Table 45: Supported Message Groups—Root Complex Mode (Continued)

Message Group	Supported	Action
Vendor Specific Messages	No	

Table 45 lists the message groups that are supported in Endpoint mode:

Table 46: Supported Message Groups—Endpoint Mode

Message Group	Supported	Action
Interrupt Signaling	Yes	Interrupt assertion on internal interface is forwarded as an Interrupt Assert message to the PCI Express port. Interrupt de-assertion on internal interface is forwarded as an Interrupt De-assert message to the PCI Express port. Both INTA, INTB, INTC and INTD are supported.
Power Management	Yes	Received PME_Turn_Off message triggers the relevant power management procedure. It is acknowledged by a PME_TO_Ack message that is generated and transmitted on the PCI Express port. PM_Active_State_Nak and PM_PME messages are not supported.
Error Signaling	Yes	Error in the PCI Express port is forwarded as an Error message to the PCI Express port. Correctable, Non-fatal, and Fatal error messages are supported.
Hot Plug Signaling	No	
Locked Transaction Support	No	
Slot Power Limit Support	Yes	When Set_Slot_Power_Limit message is received, the Slot Power Limit Configuration registers are updated accordingly.
Vendor Specific Messages	No	

Copyright © 2015 Marvell

17.13 Message Signaled Interrupt (MSI) Support

MSI interrupt support is required for PCI Express devices, and is driven by completing a memory write TLP.

MSI Root Complex (RC) Mode Support

The MSI data must be monitored in the predefined address by the software.

Upon receiving an MSI, an interrupt is set in the interrupt main cause register, and interrupt data is saved in the PCI Express MSI Message Data Register (Table 406 p. 595).

MSI Endpoint (EP) Mode Support

When enabled through the PCI Express MSI Message Control Register (Table 403 p. 594), the EP generates an MSI write for any edge interrupt assertion. The write address is set according to the PCI Express MSI Message Address Register.

For 64-bit addresses, the PCI Express MSI Message Address (High) Register (Table 405 p. 594) data content is the same as the PCI Express MSI Message Data Register (Table 406 p. 595).

17.14 Locked Transactions

Locked transaction semantics are not supported. This includes MRdLk, CplLk, and Unlock messages.

17.15 Arbitration and Ordering

The arbitration scheme on both Tx and Rx directions are following the PCI Express ordering rules. On each direction, there are separate queues for posted, non-posted, and completion TLPs that are used to adhere to these ordering rules.

17.15.1 Tx Ordering Rules

- All TLPs from same type (P, NP, C) are forwarded in-order.
- Non-Posted transactions push posted transactions (unless the <TxNpPushDis> field in the PCI Express TL Control Register (Table 442 p. 622) is set to 1).
- Completions push posted transactions (unless the <TxCmplPushDis> field in the PCI Express TL Control Register (Table 442 p. 622) is set to 1).
- For other transaction couples, re-ordering may occur.
- Relaxed-ordering is not used for ordering in internal queues.
- Simple round-robin arbitration is performed on all transactions which may be transmitted to the PCI Express fabric according to those rules.

17.15.2 Rx Ordering Rules

- All TLPs from same type (P, NP, C) are forwarded in-order.
- Non-Posted transactions push posted transactions (unless <RxNpPushDis> field in the PCI Express TL Control Register (Table 442 p. 622) is set to 1).
- Completions push posted transactions (unless the <RxCmplPushDis> field in the PCI Express TL Control Register (Table 442 p. 622) is set to 1).
- For other transaction couples, re-ordering may occur.
- Relaxed-ordering is not used for ordering in internal queues.
- Simple round-robin arbitration is performed on all transactions.

Copyright © 2015 Marvell

17.16 PCI Express Register Access

The PCI Express registers can be accessed from an external PCI Express device, or from the core processor.

The Read Only (RO) registers have the following access permissions:

- An external PCI Express device can only have read access to these registers.
- If not hardwired or set/clear by the hardware, these registers are read/write (RW) from the core processor.

If the device is configured as an endpoint, the configuration header registers can be accessed from an external Root Complex port via type0 configuration transactions.

The configuration header registers are also mapped to the chip internal address space as follows:

- All configuration header registers are mapped to the internal memory space.
- Direct memory access is enabled via the <CfgMapTo MemEn> field in the PCI Express Control Register (Table 429 p. 615). When enabled, a direct memory access can be performed from the core processor or from an external PCI Express device, even if the device is configured as Root Complex port.
- If not hardwired or set/clear by the hardware, all RO configuration header registers are RW, if accessed through the direct memory mapping.

Note If Port0 is set to a quad x1 configuration, an external PCI Express device attached to Port 0 has no access to the registers of the other ports.

17.16.1 $D3_{hot}$ to D0 Transition—Endpoint Mode

Switching from *$D3_{hot}$* state to *D0* state is done by writing to <PMState> field in the PCI Express Power Management Control and Status Register (Table 402 p. 593).

When such transition occurs, the configuration headers are reset to default values, except the following registers:

- PCI Express Device and Vendor ID Register (Table 387 p. 585)
- PCI Express Class Code and Revision ID Register (Table 389 p. 588)
- PCI Express Subsystem Device and Vendor ID Register (Table 397 p. 590)

17.17 Hot Reset

Hot Reset is an in-band reset indication that can be sent from the root-complex downstream and reset the PCI Express hierarchy. Use the following procedure to generate a hot reset:

1. Write to the <ConfMstrHot Reset> field in the PCI Express Control Register (Table 429 p. 614).
2. To check that Hot Reset has been completed, poll the <DLDown> field in the PCI Express Status Register (Table 430 p. 616). When this bit is set, DL is down and Hot Reset has been completed.
3. Clear the <ConfMstrHot Reset> field.

Note Root Complex registers are not reset by Hot Reset.

Copyright © 2015 Marvell

17.18 Error Handling

17.18.1 Physical Layer Errors

Table 47 lists the conditions that can cause a PHY layer Receive error.

Table 47: Physical Layer Error List

Error Name	Conditions
Receiver Error	• PHY Overflow • PHY Underrun • PHY 8B/10B decode error • PHY Disparity error Severity: Correctable.

17.18.2 Data Link Layer Errors

Table 48 lists the Data Link layer errors.

Table 48: Data Link Layer Error List

Error Name	Conditions
Bad TLP	• LCRC Error detected in received TLP. • Sequence number error detected in received TLP. Severity: Correctable.
Bad DLLP	CRC Error detected in received DLLP. Severity: Correctable.
Replay Timeout Error	Replay timer expired. Severity: Correctable.
REPLAY_NUM Rollover Error	REPLAY_NUM rolled-over. Four consecutive replays were transmitted. Severity: Correctable.
Data Link Layer Protocol Error	Reception of an Ack with out-of-range ackNac_Seq_Num. Severity: Fatal.

Copyright © 2015 Marvell

17.18.3 Transaction Layer Errors

Table 49 lists the Transaction layer errors.

Table 49: Transaction Layer Error List

Error Name	Description
Flow Control Protocol Error	• DLLP receive timer expiration. • Received FC initial credit values, which are less than the minimum advertisement according to the specifications. • Received update FC message for a credit type, which was advertised as infinite on initialization. Default severity: Fatal.
Malformed TLP	• Received TLP with data payload size larger than the Maximum Payload Size. • Received TLP with undefined *Type* and *Fmt* fields value. • Received TLP with length different than expected according to the length, type, and TD (TLP Digest) field. • Received request with Address/Length combination crossing the 4-KB boundary. • Received Power Management Set_Slot_Power, Unlock, INTx, and error message with TC field not equal to 0 (TC0). Default severity: Fatal.
Poisoned TLP Received.	Poisoned TLP received. Default severity: Non-Fatal.
Unsupported Request	• Received unsupported TLP type (CfgWr1, CfgRd1, MrdLk). • Received unsupported message codes. • Failed address decoding on received TLP. • Received CfgWr0 or CfgRd0 with function_number different than 0. • Received poisoned write request to internal register space. Default severity: Non-Fatal. **NOTE:** Reception of Vendor_Defined_Type_1 message is discarded silently. It is not an error state.
Received UR Completion	• Received Cpl TLP with UR completion status. • Received CplLk or CplD with UR completion status. Not a PCI Express error. Mapped to PCI status.
Completion Timeout	Outstanding Non Posted request to PCI Express timeout has expired. Default severity: Non-Fatal.
Completer Abort	Received read requests to the internal address space, with the Length field different than 1 DWORD. Default severity: Non-Fatal.
Received CA completion	Received a Cpl with CA completion status Not a PCI Express error. Mapped to PCI status.
Unexpected Completion	• Received unexpected completion TLP (Cpl or CplD). Completion does not correspond to one of the outstanding NP requests. • Received CplLk or CplDLk TLPs. Default severity: Non-Fatal.

Copyright © 2015 Marvell

17.18.4 Error Propagation

The PCI Express specification defines a mechanism for propagation of erroneous TLPs (erroneous data payload) via an EP (Error Poisoning) bit in the packet header.

Receive

Upon receiving a poisoned write TLP:

- If it is not an access to the chip internal registers and if the <RxDPPropEn> field in the PCI Express Mbus Adapter Control Register (Table 445 p. 626) is set, an erroneous data indication is forwarded along with the data. Regardless of <RxDPPropEn> field setting, Error status bits in the PCI Express configuration header registers are set (if enabled by the relevant configuration registers control bits) and Error messages are sent (if enabled).
- If it is an access to the chip internal registers (whether PCI Express register file or another register file), the transaction is discarded (not written to registers). An unsupported requests completion is sent if needed, and the relevant error status bits are set.

Upon receiving a poisoned completion TLP, if the <RxDPPropEn> field is set, an erroneous data indication is forwarded along with the data. Regardless of <RxDPPropEn> field setting, error status field are set, if enabled by the relevant control bits.

Transmit

Tx error forwarding is controlled by the <TxDPPropEn> field in the PCI Express Mbus Adapter Control Register (Table 445 p. 625). For either requests or responses, the corresponding TLP is poisoned (EP bit is set) if:

- Data received from the Mbus is marked as erroneous.
- Forwarding is enabled by the <TxDPPropEn> field.

17.18.5 Completion Timeout

The Completion timeout (Cpl TO) mechanism is defined in the PCI Express specification. When a device issues a NP request on the PCI Express port and does not receive all the related completions of the request after a pre-defined period of time, the device must indicate a Completion Timeout error status.

The Cpl TO period is set by the <Reserved> field in the PCI Express Completion Timeout Register (Table 433 p. 617). The Cpl TO mechanism can also be disabled through this register field.

If the Cpl TO expires before a Tx NP request is completed (not all completion fragments arrived):

- The relevant error status bits are set.
- An error message is transmitted, if not masked.
- The timed-out requests are completed on the Mbus with dummy read data, and with an erroneous data indication.

Copyright © 2015 Marvell

17.19 Power Management

The following sections describe the PCI Express interface power management features.

17.19.1 D1, D2, and D3 Software Power Management Options

The device supports the PCI Express software power management options D1, D2, and D3, as described in the PCI-E 2.0 Specification.

As an endpoint, it also supports the turnoff process. Upon receiving turnoff message from the Root Complex, a maskable interrupt is set. The device CPU acknowledges the turnoff request by setting the <Send Turn Off Ack Msg> field in the PCI Express Power Management Register (Table 435 p. 618).

As a Root Complex, the 88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828 can set the device it works with to a turnoff state by sending a turnoff message. To send the turnoff message, set the <Send Turn Off Msg> field in the PCI Express Power Management Register (Table 435 p. 618). When the turnoff process completes, a maskable interrupt is set in the main cause register.

As a Root complex, the 88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828 responds to a Power Management Event (PME) generated by the device. A maskable interrupt is set upon receiving a PME message. The PME data is saved in the PCI Express Root Complex Power Management Event (PME) Register (Table 434 p. 617).

17.19.2 Active State Power Management (ASPM)

An active power management event is a mechanism defined by the PCI Express specification. It allows the hardware to lower the link power state. This device supports all ASPM options as a root complex and endpoint.

The PCI Express driver should enable this feature, as described in the PCI Express specification.

L1 ASPM Endpoint Mode If this feature is enabled, the device starts the L1 ASPM event when the conditions for this event are met, as defined in the PCI Express Specification, and when the <L1 ASPM En> field in the PCI Express Power Management Register (Table 435 p. 618) is set.

L1 ASPM Root Complex Mode As a root complex, the device acknowledges an L1 ASPM event when the <L1 ASPM Ack> field in the PCI Express Power Management Register (Table 435 p. 618) is set. If this field is not set, an L1 ASPM non-acknowledge response is sent upon the L1 ASPM request.

L0 Support The device supports L0 events on receive and transmit, as defined in the PCI Express specification.

17.19.3 CLKREQn Support

The device supports dynamic gating of the output reference clock for additional power saving via CLKREQn signaling as defined in the PCI Express specification.

17.19.4 PHY Shutdown

Shutting down the PHY is done as part of the unit power down procedure.

To power down the unit, access the PCI Express Power Management Control and Status Register (Table 402 p. 592), and clear the relevant port's power up register.

Power down a port only after the software has confirmed that the link was not negotiated for at least 100 ms.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Before powering down the port, the software must clear the PCI Express enable bit for the relevant port in the SoC Control 1 Register (Table 1888 p. 1394).

Note

When working in quad mode, always supply power to port x.0, unless it is desired to power down all of the PCIe ports.

17.20 Loopback Modes

The following DFT features are supported by the PCI Express port:

- Master Loopback
- Internal Loopback
- Slave Loopback
- PRBS (Pseudo-Random Bit Sequence) generation and checking

17.20.1 Master Loopback

Master Loopback mode forces the device on the other side of the PCI Express link to enter a Loopback mode and mirror all the received traffic back to its Tx side. This mode enables a self-test procedure for the PCI Express port and link.

Only the PCI Express TS1 Order Sets are sent during Master Loopback. To check the physical connectivity of the link, monitor the receiver errors.

Entering master loopback must be done before the PCI Express link is enabled. Use the following procedure:

1. Set the <ConfMstrLb> field in the PCI Express Control Register (Table 429 p. 614).
2. Set the <PCIe0En>, <PCIe1En>, <PCIe2En>, or <PCIe3En> field in the SoC Control 1 Register (Table 1888 p. 1395) to enable the specific PCI Express port.
3. Check that the PCI Express link is ready for a loopback test by polling bit[26] in the <TrnCntlBits> field in the PCI Express Debug MAC Status 1 Register (Table 441 p. 621). When this bit is cleared, the link is in master loopback mode and the loopback test can start.
4. Check for Receiver errors.

To exit Master Loopback mode, reset the entire chip.

Copyright © 2015 Marvell

17.20.2 Internal Loopback

When working in Internal Loopback mode, the PCI Express port mirrors all the traffic received from the Mbus back to its Rx side. This mode enables self-test procedures for the PCI Express port, even when no external device is attached.

The PCI Express PHY supports Shallow and Deep Loopback modes as shown in Figure 55 and Figure 56.

Figure 55: Shallow Internal Loopback

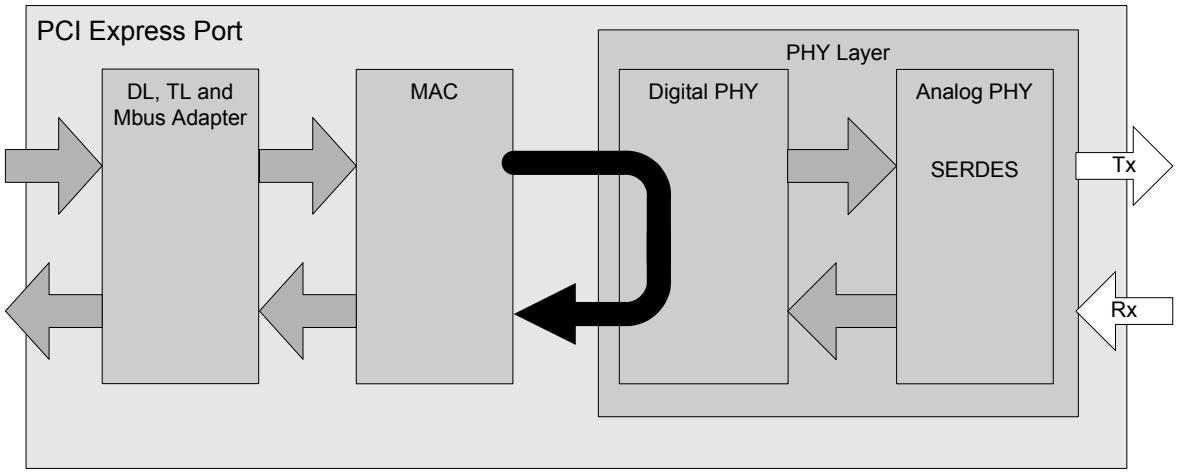
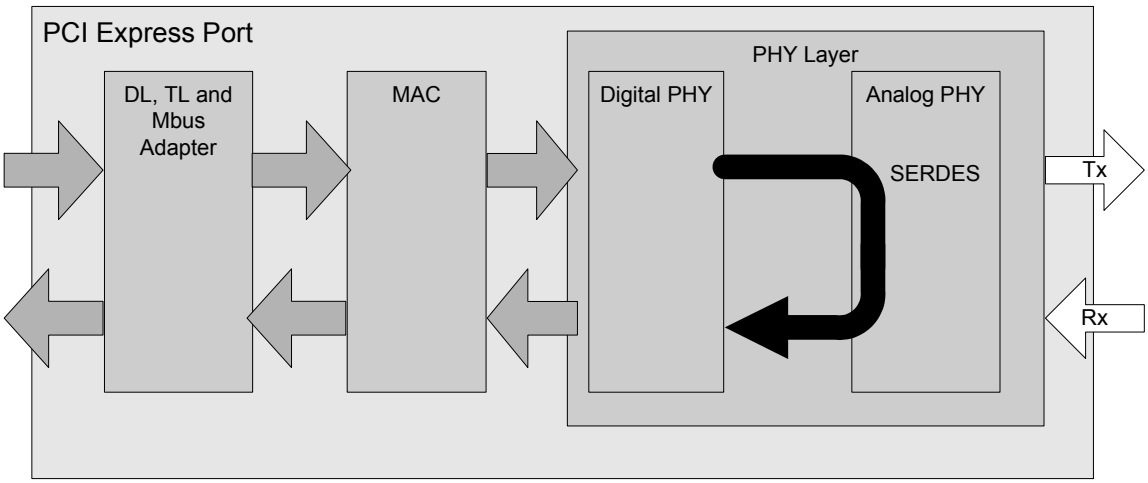


Figure 56: Deep Internal Loopback



Use the following procedure to enter Internal Loopback mode:

1. Set the following registers for Shallow, Deep, or SERDES loopback:
   - For Shallow loopback, set PCI Express PHY Register 0xC2 bit[10] to 1, using the PHY registers indirect access, as explained in Section 17.17, Hot Reset, on page 216.
   - For Deep loopback set the PHY register 0xC2 bit[11] to 1.

Copyright © 2015 Marvell

- For SERDES loopback, set the PHY register 0xC2 bit[12] to 1.

2. Set PCI Express PHY register 0x23 bits[13] to 0x1.

   NOTE: The above step does not apply for Shallow loopback.

3. Set the <PCIe0En>, <PCIe1En>, <PCIe2En>, or <PCIe3En> field in the SoC Control 1 Register (Table 1888 p. 1395) to enable the PCIe ports.
4. Check that the PCI Express Link is ready for the loopback test by polling the <DLDown> field in the PCI Express Status Register (Table 430 p. 616). When this bit is cleared, loopback test can start.
5. Set the relevant registers to control the inbound traffic (BAR, address space control and the address decoding windows).
6. Generate the loopback test traffic and check that it is received correctly.



Note

Deep Internal loopback requires a 50 ohm terminations on the transmitter signals, and the link should be in compliance mode.

Set the link to compliance mode by either terminating the Tx signals, or by forcing Rx present by setting bits[15:14] of register 0x180 to 0x2 (force Rx present).

To exit Internal Loopback mode, generate hot reset or reset the entire chip.

17.20.3 PRBS

When using an internal loopback, it is possible to use PHY PRBS generation and checking, rather than activate the entire chip to generate traffic.

To enable PRBS generation, choose the pattern by configuring PHY registers 0x15 bits[7:4], set PHY register 0x15 bit[15] to 1. PRBS errors are counted in PHY registers 0x20 and 0x1F. To reset the PRBS error counter, set PHY register 0x15 bit[14] to 1.

17.20.4 Slave Loopback

Slave Loopback is the opposite case of Master loopback. This procedure is initiated and controlled by the external PCI Express device.

17.21 Peer-to-Peer Traffic

The device supports the memory transactions between the PCI Express port 0 and PCI Express ports 1-3.

The device does not comply with the PCI transparent bridge specification:

- It does not implement a PCI bridge configuration header.
- It does not support type1 configuration cycles forwarding.
- It does not comply with the errors forwarding specification.

However, it is still very useful as a non-transparent bridge:

- It supports forwarding of memory and I/O transactions.
- It supports forwarding of PCI interrupts to PCI Express interrupt messages.
- It supports configuration cycle forwarding via indirect access, using the PCI Express Configuration Address Register and the PCI Express Configuration Data Register.

Copyright © 2015 Marvell

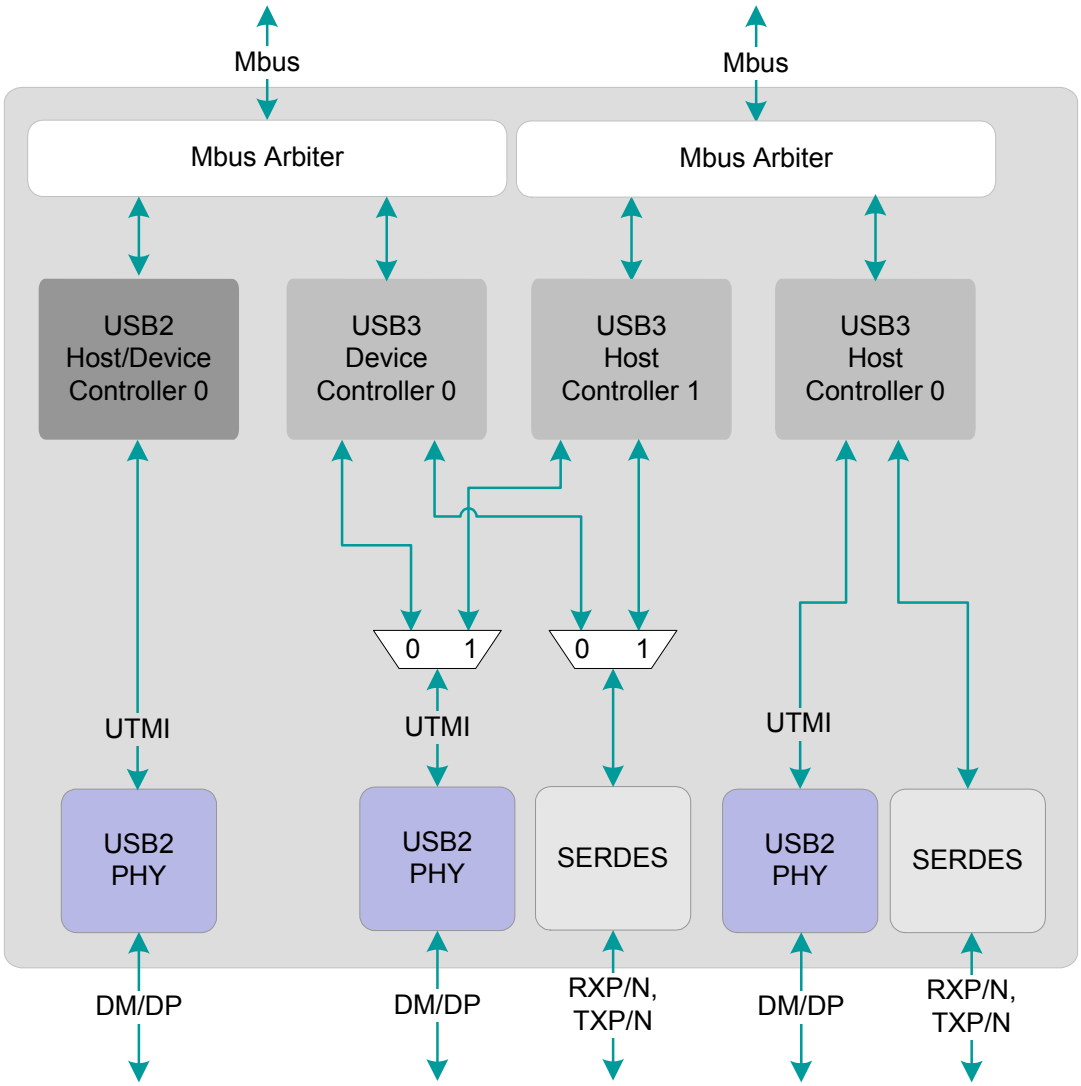
18 Universal Serial Bus (USB) Controller

This section describes the USB 3.0 interfaces integrated into the device. Even though there are 4 USB controllers, 3 USB 3.0/2.0 PHY ports are supported.

- 1x USB 2.0 Host/Device controller
- 2x USB 3.0 Host controller
- 1x USB 3.0 Device controller

The block diagram in Figure 57 shows the different USB controllers connection to the PHY interfaces integrated into the device.

Figure 57: USB 3.0 Host and Device Controller Block Diagram



Copyright © 2015 Marvell

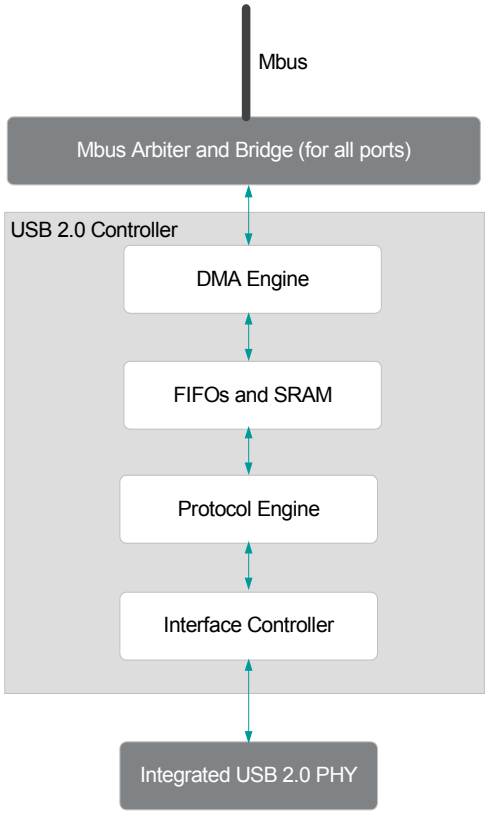
Note The full matrix of SERDES configuration options for this device is included in the *High-Speed SERDES Multiplexing* section and the *SERDES Lane Selection Options* table in the device Hardware Specifications.

The USB Configuration Register (Table 1906 p. 1408) controls this USB configuration scheme. The implementation sequence is controlled by the U-Boot.

18.1 Universal Serial Bus (USB 2.0) Interface

The port integrates a USB 2.0 host and device controller (including a DMA and protocol engines) and an embedded USB 2.0 compatible PHY.

Figure 58: USB 2.0 Port Block Diagram



The USB 2.0 registers are located in Appendix A.9, USB 2.0 Registers, on page 691 and Appendix A.10, USB2 PHY (UTMI PHY) Registers.

Note For more details about the USB 2.0 controller, refer to the http://www.usb.org/developers/usb20/.

18.1.1 USB 2.0 Controller Features

- USB 2.0 compatible
- Enhanced Host Controller interface (EHCI) compatible as a host
- As a host, supports direct connection to all peripheral types (LS, FS, HS)
- As a device, connects to all host types (HS, FS) and hubs
- 6 independent end points supporting control, interrupt, bulk, and isochronous data transfers
- Wake on USB options

18.1.2 USB 2.0 PHY Features

- Embedded USB 2.0 compatible PHY
- 480 Mbps High Speed (HS), 12 Mbps Full Speed (FS) and 1.5 Mbps Low Speed (LS) serial data transmission rates
- Synchronization/End-of-Packet (SYNC/EOP) generation and checking
- Data and clock recovery from serial stream on the USB
- Non Return to Zero Invert (NRZI) encoding/decoding with bit stuffing/unstuffing
- Bit stuffing/unstuffing; bit stuff error detection
- Frame error detection
- Holding registers to stage transmit and receive data
- Low power suspend mode
- Resume signaling
- Wake up and Suspend detection
- Supports USB 2.0 Test modes
- Ability to switch between FS and HS terminations/signaling

18.1.3 Power Management

When the USB port is placed in suspend low power state, the PHY power consumption can be minimized by turning off blocks that are not required for the suspend/resume operation. To enable this, set the <PHCD> field in the USB2_PORTSC Register (Table 1249 p. 1025) to 1.

The port wakes up from the low power mode and resumes operation when configured to do so by the host software.

Unused USB ports may be forced into a suspend low power state as follows:

1. Set <Enable force suspend> field in the USB 2.0 PHY Suspend/Resume Control Register (Table 559 p. 697) to 1 to enable forced suspend.
2. Clear <SUSPENDM> field in the USB 2.0 Power Control Register (Table 558 p. 696) to 0 to force PHY suspend.

When the ARMADA® 38x Family SoC is in Idle or Deep Idle low power modes, the USB port can wake the SoC with one of the following maskable events:

- Resume signaling by a USB device
- USB device attachment
- USB device detachment

Copyright © 2015 Marvell

18.2 Universal Serial Bus (USB) 3.0 Host Controller

This section describes the features of the USB 3.0 Host interface, including a high-level block diagram and a description of the blocks in the diagram.

The USB3 host controller provides highly-powerful efficient operation, higher performance, and extensibility to support the USB 3.0 specification. This host controller is compliant with the eXtensible Host Controller Interface (xHCI) that ultimately replaces UHCI, OHCI, and EHCI[1] and provides an easy path for new USB specification and technologies.

The host controller supports all USB respective speeds including USB 3.0 SuperSpeed (SS) and USB 2.0 HS/FS/LS.

For the USB 3.0 Host registers, refer to the *eXtensible Host Controller Interface for Universal Serial Bus (xHCI)* specification, Revision 1.0.

The USB 3.0 Host Controller Bridge registers are located in Appendix A.12, USB 3.0 Host Controller Bridge Registers, on page 745.

18.2.1 Features

The USB3 host controller integrates the following features:

- 64 bits AXI master system bus interface
  - 1 AXI master
  - 8 outstanding read requests and 8 outstanding write requests
- 32-bit AHB slave register programming interface
- 32 device slots
- 1 SuperSpeed root hub port and 1 USB 2.0 root hub port
- 1 Bus Instance
- 1 interrupt target
- Up to 64 event ring segment table
- Descriptor caching and prefetching for predictable performance
- 64 Endpoints (EP) in SuperSpeed periodic PIPE, and 64 EPs in USB 2.0 periodic PIPE
- 64 EPs in SuperSpeed non-periodic PIPE, and 64 EPs in USB 2.0 non-periodic PIPE
- Separate SuperSpeed and USB 2.0 transaction transfer
- 16 KB for RX buffer size, and 4 KB for TX buffer size
- 32-bit addressing
- Scatter/gather to minimize application level data copies
- Link U1, U2, U3 power management feature
- Supports features specified in USB 3.0 specification
- Supports features specified in USB 2.0 specification for HS/FS/LS
- 16-bit/250-MHz 32-bit/125-MHz PIPE interface to PHY
- 8-bit/60-MHz or 16-bits/30-MHz UTMI interface

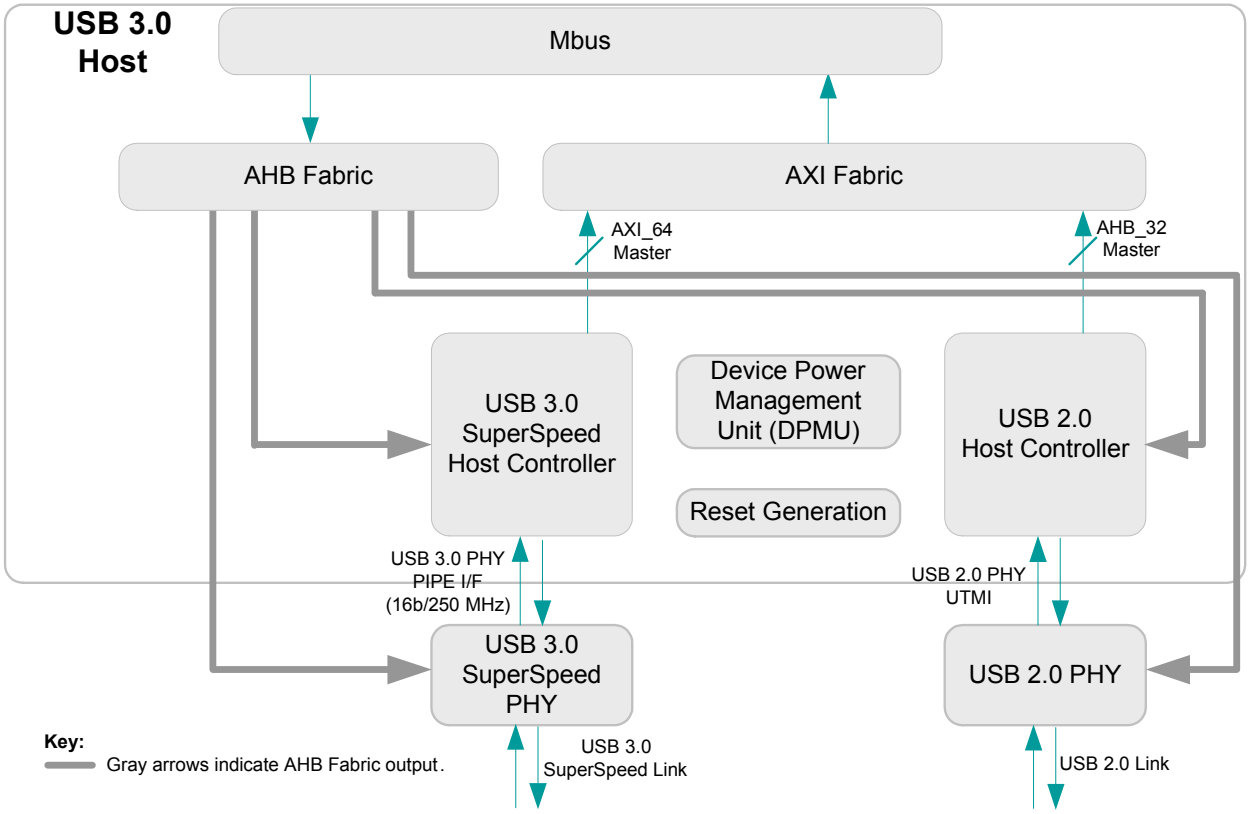
1. UHCI = Universal Host Controller Interface, OHCI = Open Host Controller Interface, and EHCI = Enhanced Host Controller Interface

Copyright © 2015 Marvell

18.2.2 Host Controller Top Level Diagram

Figure 59 shows the top-level structure of the USB 3.0 host controller. On the system interface side, the data bus uses a 64-bit AXI bus. The register access bus uses 32-bit AHB bus. Both the AXI and AHB buses are connected to the Mbus.

Figure 59: USB 3.0 Host Top Level Diagram



18.2.3 Operational Model and Register Interface

The USB 3.0 Host controller operational model and register file is compliant to the eXtensible Host Controller Interface (xHCI) specification revision 1.0. Refer to this specification for more information.

Copyright © 2015 Marvell

18.3 Universal Serial Bus (USB) 3.0 Device Controller

The USB 3.0 device controller supports SuperSpeed (SS), High Speed (HS) and Full/Low Speed (FS/LS) function and it is backward compatible with USB 2.0. The USB 3.0 SuperSpeed has a fast, bidirectional bus interface running at a 5 Gbps rate in each direction. The USB 3.0 HS or FS/LS has a shared bus interface and runs at 480 Mbps, 12 Mbps, and 1.5 Mbps, respectively.

Figure 57, USB 3.0 Host and Device Controller Block Diagram, on page 224 shows the top-level structure of the USB 3.0 host and device controller.

The USB 3.0 Device registers are located in Appendix A.13, USB 3.0 Device Registers, on page 748.

18.3.1 Features

SuperSpeed Feature Highlight

- 64 bits AXI master system bus interface. Supports 8 outstanding read requests and 8 outstanding write requests.
- 32 bits AHB slave register programming interface.
- 15 IN Endpoints, 15 OUT Endpoints and 1 control Endpoint
- 4 types of transfers:
  - Control
  - Bulk (Bulk only and Bulk stream)
  - Interrupt
  - Isochronous
- Hardware accelerated Bulk stream engine to maximize the performance
- Support maximum packet size for each Endpoint (EP).
- For IN EP, a 2 KB data buffer per EP. All OUT EPs share an 8 KB data buffer.
- Link layer header flow control.
- Power management and remote wake up
- 16 bits/250 MHz or 32 bits/125 MHz PIPE interface to PHY

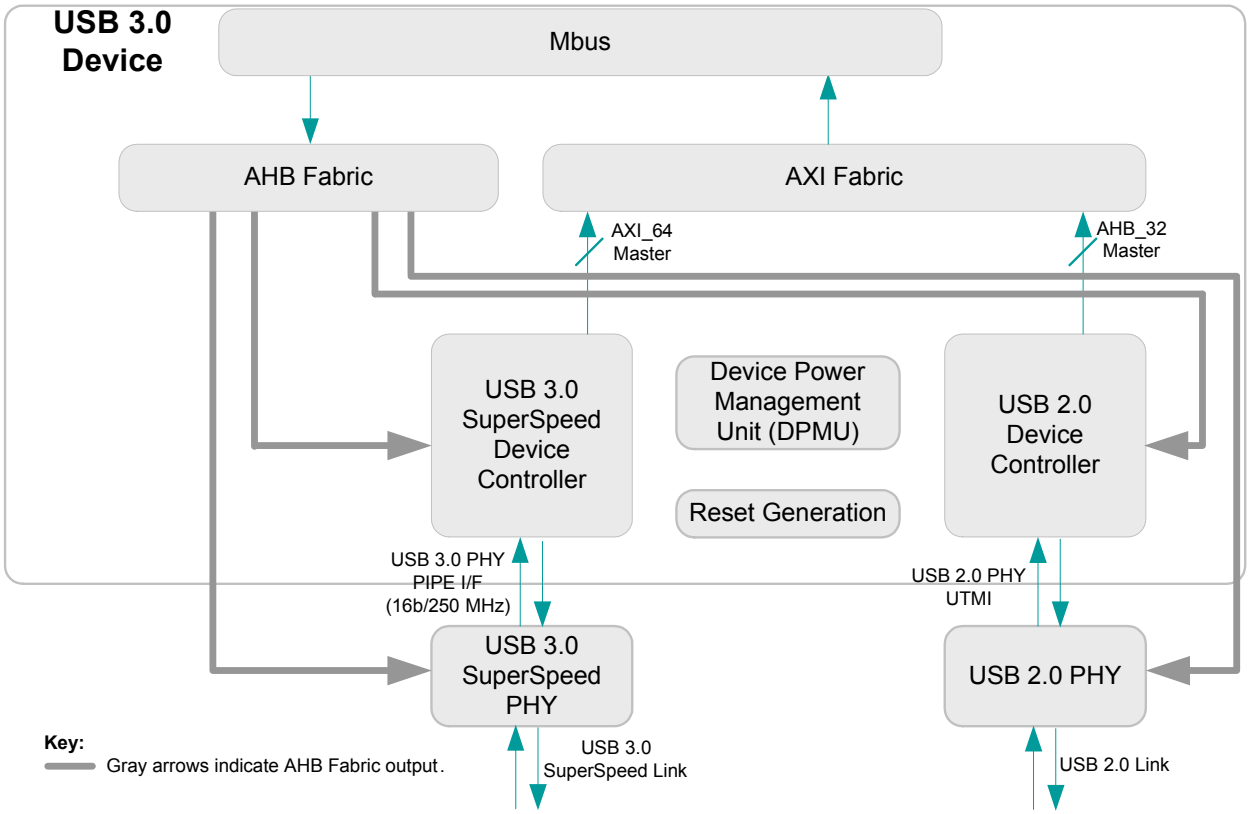
HS, FS/LS Feature Highlight

- 64 bits AXI master system bus interface
- 32 bits AHB slave register programming interface
- Support all features in USB 2.0 specifications
- 8 bits/60 MHz or 16 bits/30 MHz UTMI interface to PHY

Figure 60 shows the top structure of USB 3.0 device controller. On the system interface side, the data bus uses a 64-bit AXI bus. The register access bus uses a 32-bit AHB bus. Both the AXI and AHB buses are connected to the Mbus.

Copyright © 2015 Marvell

Figure 60: USB 3.0 Device Top Level Diagram



18.3.2 Reset Scheme

The device applies system reset and software reset to the USB 3.0 subsystem. The USB 3.0 host initiates warm or hot reset to reset USB 3.0 SuperSpeed, and bus reset to reset USB 2.0. The mechanism the IP provided can also support other reset sources if required by application. For inband reset like warm reset and hot reset, Table 50, Reset on Device SuperSpeed Logic Module and State, on page 231, indicates the reset impact on each logic module and device states.

The following describes different reset sources to the USB 3.0 subsystem.

- **System Reset:** This is SoC level reset. It can be power on reset or on any other SoC level reset. It is input to the DPMU block. The DPMU generates several reset outputs for different clock domains and modules. The system reset should reset the entire device, including the DPMU block.
- **SuperSpeed Software Reset:** This is programmed through SuperSpeed configuration registers. For the SuperSpeed core, this reset has the same impact as power on reset.
- **Warm Reset:** This is host initiated reset when the device is connected to the SuperSpeed interface. Once this reset is received, the interconnect master module and slave modules need to terminate their processes in a clean manner on the interconnect bus. The SuperSpeed core resets most of its internal modules, and it then transitions the device state to the Powered state. The following actions are also performed based on the USB 3.0 specification:
  - Reset of port configuration information.
  - Re-initialization of Rx equalization and other PHY related parameters.
  - LTSSM (Link Training Status State Machine) transition to RX.Detect state.

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

- **Hot Reset:** This reset is from the host when the device is connected on the SuperSpeed interface. It is similar to the warm reset in terms of the requirement for the application layer. For the link layer, protocol layer and PHY, the requirement is slightly different from the warm reset. The hot reset is required to performs the following actions:
  - Port configuration information will not be reset.
  - Rx equalization and other PHY related parameters should not be re-initialized.
  - The LTSSM should transition to Hot Reset.
- **USB 2.0 Software Reset:** This reset is programmed in the USB 2.0 core configuration registers. The reset is processed inside USB 2.0 core.
- **USB 2.0 Bus Reset:** This reset is from the USB Host controller to reset the USB 2.0 device.

Table 50 provides the reset on device SuperSpeed logic module and state for the first 4 reset sources described above.

Table 50: Reset on Device SuperSpeed Logic Module and State

	System Reset (Power on Reset)	SuperSpeed Software Reset	Warm Reset (HW)	Hot Reset (HW)
Device State	Transition to Powered state. (Far End Receiver Termination)	Transition to Powered state. (Far End Receiver Termination)	Transition to Powered state. (Far End Receiver Termination)	Transition to Default state.
Device Port State	Transition to Powered on state.	Transition to Powered on state.	Transition to Powered on state.	Stay at SS_CONNECTED state.
PHY	PHY level variables should be reset to power-on default value. PLL should be reset to power-on default state. SuperSpeed receiver and transmitter are not ready. LFPS receiver and transmitter are not ready. RX Termination off.	Same as power-on reset.	RX Termination on SuperSpeed receiver and transmitter are ready. LFPS receiver and transmitter are ready. PIPE clock and core clock are on.	RX Termination on. SuperSpeed receiver and transmitter are ready. LFPS receiver and transmitter are ready. PIPE clock and core clock are on.
Link MAC	All signals are reset to power-on default value. All state machines are reset to power-on default state. LTSSM after reset is in Start state, in which RX termination is 0 and power state is P2.	Same as power-on reset except LTSSM should be in SuperSpeed. Disable (P3) state after soft reset is finished. Once softConnect is on, the LTSSM should transition to RX.Detect.Reset	The port LTSSM should transition to RX.Detect. The PIPE interface logic should not be reset. The data path state machines should be reset to IDLE state.	The port LTSSM should transition to Hot Reset. The PIPE interface logic should not be reset. The data path state machines should reset to IDLE state. Scrambling should keep intact.

Copyright © 2015 Marvell

Table 50: Reset on Device SuperSpeed Logic Module and State (Continued)

	System Reset (Power on Reset)	SuperSpeed Software Reset	Warm Reset (HW)	Hot Reset (HW)
Link Data	All signals are reset to power-on default value. All state machines are reset to power-on default state. After reset is completed, the LINK CREDIT is 4, and the LINK SEQ Number is 0.	All signals are reset to power-on default value. All state machines are reset to power-on default state. After reset is completed, the LINK CREDIT is 4, and the LINK SEQ Number is 0.	All signals reset to power-on default value, and state machines go to IDLE state. The LINK CREDIT goes to 4, and the LINK SEQ Number to 0.	All signals reset to power-on default value, and state machines go to IDLE state. The LINK CREDIT goes to 4, and the LINK SEQ Number to 0.
Protocol	All signals are reset to power-on default value. All state machines are reset to power-on default state. After reset is completed: • EP0 is able to transmit and receive. • Nonzero EPs are disabled. • The Sequence number for all EPs is reset to 0.	All signals are reset to power-on default value. All state machines are reset to power-on default state After reset is completed: • EP0 is able to transmit and receive. • Nonzero EPs are disabled. • The Sequence number for all EPs is reset to 0.	All signals reset to power-on default value, and state machines go to IDLE state. EP0 is able to transmit and receive. Nonzero EPs are disabled. The Sequence number for all EPs is reset to 0.	All signals reset to power-on default value, and state machines go to IDLE state. EP0 is able to transmit and receive. Nonzero EPs are disabled. The Sequence number for all EPs is reset to 0.
DMA	Flush FIFO Reset DMA state machines.	Flush FIFO Reset DMA state machines. Notify SW when warm reset is completed.	Flush FIFO Reset DMA state machines Notify SW when warm reset is completed.	Flush FIFO Reset DMA state machines. Notify SW when warm reset is completed.
EP0_processor	Reset to Power-On default State.	Reset to Power-On default State.	Reset to Power-On default State.	Reset to Power-on default State.
Register file	All registers should reset to power-on default value.	All registers should reset to power-on default value except power management registers.	All registers should reset to power-on default value except following registers: LFPS.ping, LFPS.polling FPS.u1exit LFPS.u2exit LFPS.u3exit LFPS.warmreset Timer enable registers	All registers should reset to power-on-default value except following registers: port capability LFPS.ping, LFPS.polling FPS.u1exit LFPS.u2exit LFPS.u3exit LFPS.warmreset Timer enable registers

Document Classification: Proprietary Information

Table 50: Reset on Device SuperSpeed Logic Module and State (Continued)

	System Reset (Power on Reset)	SuperSpeed Software Reset	Warm Reset (HW)	Hot Reset (HW)
AXI Wrapper	All signals are reset to power-on default state. All state machines are reset to power-on default state.	Wait all outstanding AXI transaction to complete. AXI transaction will complete within 1 ms even if AXI runs into errors.	Wait all outstanding AXI transaction to complete. AXI transaction will complete within 1 ms even if AXI runs into errors.	Wait all outstanding AXI transaction to complete. AXI transaction will complete within 1 ms even if AXI runs into errors.
USB3 PHY	All signals are reset to the power-on default value.	All signals are reset to the power-on default value.	Not affected.	Not affected.

18.3.3 USB 3.0 SuperSpeed Core

This core performs all required functions to support features specified in USB 3.0 specification for USB 3.0 SuperSpeed. It supports USB 3.0 optional features such as Latency Tolerance Message Device Notification and function remote wake up. Those features can be enabled if required by application.

The SuperSpeed core logic operates in 3 clock domains—the SuperSpeed core clock domain, PHY PIPE clock domain and reference clock domain. The PHY PIPE clock domain and reference clock apply only to the MAC layer logic interface to the USB 3.0 PHY. All other logic in the SuperSpeed core uses the SuperSpeed core clock that typically runs at 125 MHz.

18.3.3.1 SuperSpeed DMA and Buffer Descriptor (BD)

The SuperSpeed DMA controller provides a mechanism for the user-logic to communicate endpoint specific data between SoC system memory and device local memory through the AXI system bus. The DMA interfaces with USB 3.0 device protocol layer on one side and the AXI masters on the other side.

The DMA Controller implements per endpoint FIFO for buffering USB 3.0 data payloads. The endpoint FIFOs are implemented using 2 internal SRAMs for receive and transmit, respectively. The receive buffers are shared. The transmit buffers are per EP. To support burst properly, the endpoint FIFO size should be configured to absorb the latency between system memory and local memory inside IP.

The DMA controller maintains independent DMA states for each endpoint engine. The DMA controller is used to move the USB 3.0 data payload between the endpoint FIFO and the endpoint buffers in an external system memory over the AXI interface. The endpoint buffers are described by Buffer Descriptors (BD). The Buffer Descriptor is a 4 Dword data structure that describes the size of the buffer, the address of the buffer, a pointer to the next buffer descriptor, and a few additional control fields. The endpoint engine is responsible for fetching the Buffer Descriptor from the external system memory, and updating the Buffer Descriptor completion data to the BD completion queue (called DONEQ) in system memory. This is done after it has finished processing a BD. Each endpoint engine can be independently started, stopped, resumed, and reset via DMA specific control registers accessible to the software over the AHB interface. The status of the endpoint engine is reported via DMA specific status registers.

Copyright © 2015 Marvell

DMA Buffer Descriptor Structure

A generic structure of the buffer descriptor is shown in Table 51, and the buffer descriptions appear in Table 52.

Table 51: Buffer Descriptor Structure

DWORD0	COMMAND/BUFFER SIZE
DWORD0	COMMAND/BUFFER SIZE
DWORD1	BUFFER POINTER
DWORD2	NXTBDPTR
DWORD3	DATALENGTH

The buffer descriptor is slightly different for IN and OUT transfers. The structures defined are sufficient for supporting all types of transfer mechanism defined by the USB 3.0 specification.

- **Transmit:** The Transmit Buffer Descriptor is used by IN endpoints for describing data buffers from where USB 3.0 SS payload is read and transmitted over the USB 3.0 Link.
- **Receive:** The Receive Buffer Descriptor is used by OUT endpoints for describing data buffers where USB 3.0 SS payload is received over the USB 3.0 Link.

All reserved fields should have the value of 0.

Table 52: Buffer Descriptor Field Descriptions

Bit	[Num]Symbol	Description
0:[31]	BUF_RDY	Buffer Ready: Set by the USB 3.0 function to indicate to the DMA Controller that the buffer pointed by this descriptor is ready for transmission or receive to/from the USB 3.0 host. The BD completion by DMA controller occurs: **Transmit:** 1. When the number of bytes specified by the BUF_SIZE field is transferred by the DMA Controller for non-isochronous endpoints. In case the ZLP field is set and the BUF_SIZE field is multiple of 1KB, then the completion occurs only when the final Zero Length Packet is transmitted. Note that a successful completion of transfer for non-isochronous endpoints guarantees that the data has been received by the host and the host has ACKed the transmitted data packets. 2. For isochronous endpoints when all bytes specified by the BUF_SIZE field is transferred by the DMA or when a service interval is reached. 3. Before DMA reset by SW. **Receive:** 1. If the number of bytes specified by BUF_SIZE field is transferred to the system memory. 2. If the number of bytes specified by BUF_SIZE field is not transferred but payload with a short packet is received and the corresponding payload is posted to the system memory. 3. If the number of bytes specified by BUF_SIZE field is not transferred and a zero length packet is received and all payload prior to the zero length packet have been posted to the system memory. 4. If the number of bytes specified by BUF_SIZE field is not transferred, but the payload the LPF bit is set (for ISO EP), and the corresponding payload with this packet, and all prior payloads have been posted to the system memory. 5. Not all bytes specified by BUF_SIZE are received from host when a service interval is reached. 6. Before DMA reset by SW.

Copyright © 2015 Marvell

Table 52: Buffer Descriptor Field Descriptions (Continued)

Bit	[Num]Symbol	Description
0:[30]	INT_EN	Interrupt Enable: Set by the USB function to request the DMA to generate interrupt on completing the transfer for this BD. The DMA controller will not modify this field.
0:[29]	NEXT_BD_RDY	Next BD Ready: Set by the USB Function to indicate the DMA that the next BD is read.
0:[28]	JUMP_BIT	The address of the next buffer descriptor is determined as follows: 0: The next buffer descriptor is located immediately after the current buffer descriptor. 1: The next buffer descriptor is located at the address pointed by NXTBDPTR.
0:[27]	FLUSH	Used to support Bulk Stream UAS protocol. When set, indicates the stream associated with the BD has been rejected or flow controlled by host. This bit is set by HW and used by HW. SW does not change this field or use this bit for debugging.
0:[26]	ABORT	Used to support Bulk Stream UAS protocol. When set, indicates HW should not schedule any transfer for this BD and should simply return it to SW.
0:[25]	ZLP	Used only for DMA transmit. This field indicates that a zero length packet should be transmitted after all data indicated by the BUF_SIZE is transmitted. This bit is ignored if the BUF_SIZE value is not a multiple of 1KB.
0:[24]	CHAIN	BD chain bit to indicate there are more BDs for the same transfer following this BD. The last BD in the same transfer will have this bit set 0. When receiving a short packet from the USB 3.0 host, the remaining BDs in the same transfer will be returned to system SW.
0:[23:16]	Encoded_Bulk Stream_ID	Provided by SW to support Bulk Stream. This field is reserved when EP is not configured to support Bulk Stream.
0:[15:0]	BUF_SIZE	Buffer Size in bytes written by the software to indicate the DMA total number of bytes to be read or written from/to the system memory location pointed by the buffer pointer.
1:[31:0]	BUF_POINTER	Buffer Pointer Value: Address of the external memory where the IN / OUT data resides for transmission to/from USB host.
2:[31:0]	NXTBDPTR	Next BD Buffer Pointer: The 32 bit Buffer pointer identifies where the next BD is located in the system memory when LASTBD bit is set. This needs to be aligned on a Dword boundary.
3:[15:0]	DATALENGTH	Used only when the EP is configured to support Bulk Stream. The filed is written by HW to temporally store the number of bytes transmitted or received by the DMA to/from USB 3.0 host. If EP is not Bulk Stream, the field is reserved and should be set to 0.
3:[31:16]	BULK_STREAM_ID	Bulk Stream ID when EP is configured to support bulk stream. Reserved field for other EP type.

Copyright © 2015 Marvell

BD Completion Data Structure

There is a separated memory structure located in the system memory to store the completion data for each BD. It is called DONEQ. When DMA finishes processing a BD, it updates the BD completion data to the DONEQ. The completion data is a 2-Dword data structure and each field in the DOENQ is indicated by Table 53.

The size of the DONEQ is configured by system software. The registers for start address and end address of the DONEQ are programmed by software before DMA starts.

Table 53: DONEQ Field Descriptions

Dword [Bit Num]	Symbol	Descriptions
0:[31:0]	BD_ADDR	The 32 bit Buffer pointer identifies where the BD is located in the system memory.
1:[31:16]	DATALENGTH	This field indicates the number of bytes transmitted or received by the DMA to/from USB3 host.
1:[15:8]	ISO_ERR_BIT_MAP	This field indicates which 1KB in the current BD has received data with the CRC error if MAX BUF_SIZE <= 8KB. If 8KB < BUF_SIZE <= 16KB, the bit map bit indicates which 2KB in the BD has the CRC error and so on. Used only for ISO OUT EP. For other EP types, the field is reserved and should be set to 0.
1:[7]	ISO_SEQ_NUM_ERR	This field is only used for ISO OUT EP. For other EP type, this field is reserved and should be set to 0. When set, indicates the device has received data with a sequence number error. All the packets following this error packet will be discarded in the same service interval. The DATALENGTH field indicates the total data bytes received without a sequence number error.
1:[6]	ISO_TRANS_ERR	This field is only used for ISO. For other EP type, this field is reserved and should be set to 0. **IN EP** For IN EP, this error indicates that the data bytes indicated by BUF_SIZE has not been transmitted out when the current service interval ends, due to one of the following reasons: 1. No receive host request for data or the received host request for data arrived too late for the current service interval. 2. When receive host request for data, no data is available in TXFIFO. The DATALENGTH field indicates the total data bytes transmitted before the current service interval ends. **OUT EP** For OUT EP, this error indicates that the data bytes indicated by BUF_SIZE has not been received from host when the current service interval ends, due to one of the following reasons: 1. Host has not send data or has send data arrived too late for the current service interval. 2. When receive data from host, the buffer is not available or RXFIFO is full. The DATALENGTH field indicates the total data bytes received before the current service interval ends.
1:[5]	Reserved	Value is set to 0.
1:[4]	AXI_ERROR	This error indicates the error occurred during data phase read/write on AXI bus.

Copyright © 2015 Marvell

Table 53: DONEQ Field Descriptions (Continued)

Dword [Bit Num]	Symbol	Descriptions
1:[3]	SHORT_PKT	When set, indicates a short packet is received. All following BDs that belong to the same transfer are returned to SW before scheduling any transfer. This field is reserved for transmit DMA and should be set to 0.
1:[2]	FLUSH	When set, indicates the stream ID in this BD was rejected or previously flow controlled by the host. Used for bulk stream only. For other configuration, this field is reserved and should be set to 0.
1:[1]	ABORT	When set, indicates the BD is aborted by SW stream abort request. Used for bulk stream only. For other configuration, this filed is reserved and should be set to 0.
1:[0]	CYCLE	When set, synchronizes SW and HW read/write operation on DONEQ. The cycle bit is toggled when the DONEQ FIFO write pointer wraps around.

18.3.4 USB 2.0 Controller

During the port connection process, if the link fails to connect on the SuperSpeed interface, it falls back to connect on USB 2.0.

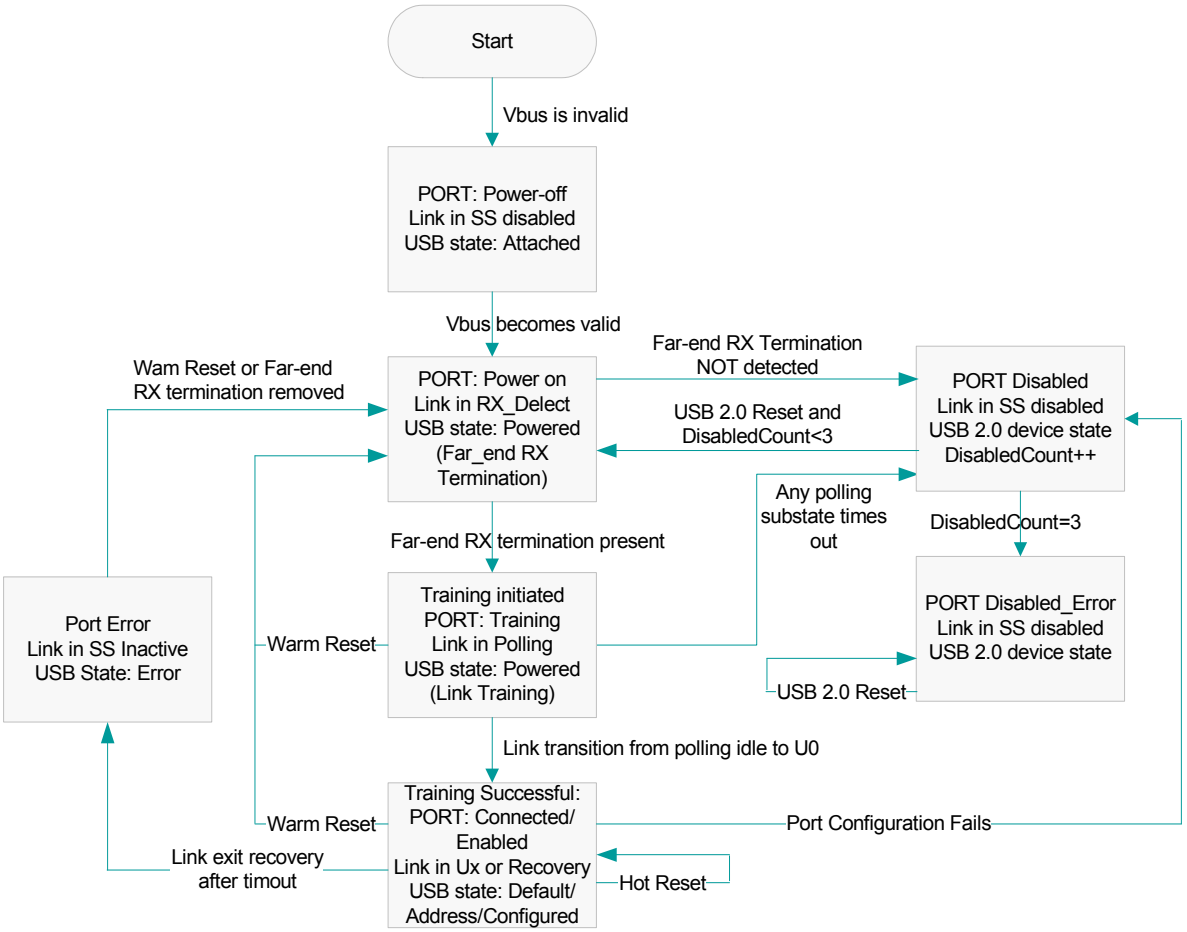
18.3.5 Device Connection State Machine

This state machine described when and how the device is connected to SuperSpeed or USB 2.0. The Vbus is provided to the USB 3.0 SuperSpeed if the device is connected to SuperSpeed (see Figure 61). Otherwise the Vbus is provided to USB 2.0 logic.

- If device is connected to USB 2.0 interface and it receives a USB 2.0 reset, the device should enter the Power On state within 1 ms.
- If the SuperSpeed port enters Port Training state, the port should disconnect on the USB 2.0 interface in 1 ms.

Copyright © 2015 Marvell

Figure 61: Device Port Connection State Machine



18.3.6 Soft Connect/Disconnect

Purpose

There are 2 usage models for this feature:

- To allow software to prepare different descriptors and BDs and to program necessary registers before device starting the enumeration process when the device is plugged in and the Vbus is on.
- To support dynamic removal and subsequent attachment to present an alternative device configuration.

Mechanism

Delay link up by controlling the Powerpresent signal from PHY to USB 3.0 core. The Powerpresent signal is an indication of Vbus availability detected by the PHY analog. A control signal is added to gate the Powerpresent so that it is only visible to the USB 3.0 LTSSM when software has finished the descriptors and register programming.

Copyright © 2015 Marvell

18.3.7 Power Management

This section offers overview of the device power management unit for USB 3.0 Device subsystem.

This section describes the:

- SuperSpeed Link power states
- USB 2.0 Link power states

18.3.7.1 SuperSpeed Link Power States

Link power management enables a link to be placed into a lower power state when the link partners are idle. The link power states are summarized in Table 54.

After been configured by software, the U1 and U2 link states are entered and exited by the hardware autonomous control, based on the link activities. Software can also put the link in low power state by programming a register control bit. The U3 link state, however, is initiated by the downstream port. U3 link is exited either by host initiated exit or device remote wakeup. The U3 link state is directly coupled to the device's suspend state.

The exit process from low power states consists of a LFPS handshake followed by link recovery and training.

Table 54: SuperSpeed Link Power States

Link State	Description	Characteristics	Typical Exit Latency Range
U0	Link active	Link operational state	N/A
U1	Link idle fast exit LFPS is supported.	Rx and Tx circuitry quiescent	µs
U2	Link idle slow exit LFPS is supported.	Clock generation circuitry may additionally be quiescent.	µs - ms
U3	Link suspend LFPS is supported.	Physical Layer power may be removed.	ms

18.3.7.2 USB 2.0 Link Power States

USB 2.0 supports the 4 link states listed in Table 55.

L0, L1 ■ The transition from L0 to L1 is controlled by the port command called LPM token sent by USB Host.

L2 ■ For a link entering L2, the device detects the suspend condition after 3 ms of idle.

L3 ■ L3 is link off state.

Table 55: USB2.0 Link Power States

Link State	Description	Characteristics	Typical Exit Latency Range
L0	Link active	Link operational state	N/A
L1	Link idle fast entry and exit	Rx and Tx circuitry quiescent	Entry: ~10 µs Exit: ~70 µs–1 ms
L2	Link idle slow entry and exit	PLL off. Physical Layer power may be removed.	Entry: ~3 ms Exit: >30 ms
L3	Off	Power off state	N/A

Copyright © 2015 Marvell

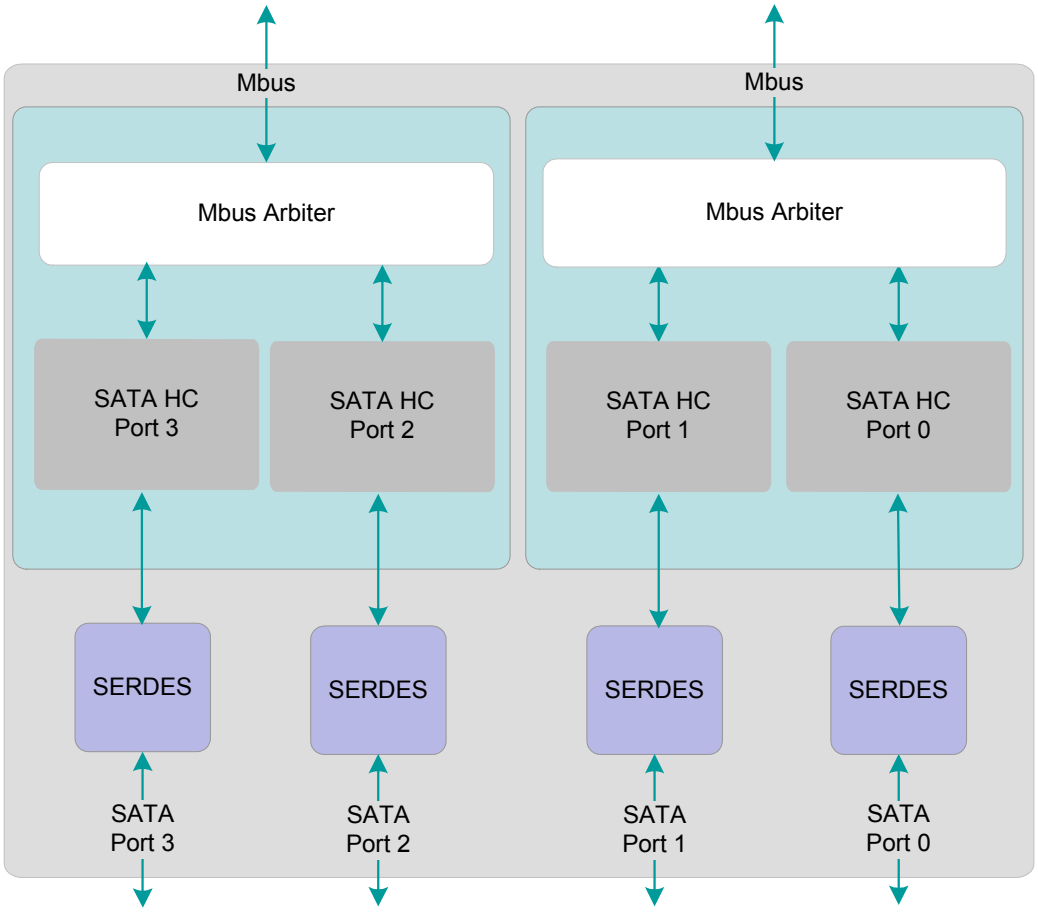
19 Serial Advanced Technology Attachment (SATA) 3.0 Host Controller

This section describes the SATA 3.0 Host controller. It includes an overview of the SATA 3.0 features and a description of the AHCI operating mode.

The device integrates SATA ports accessible to the Host through the Mbus fabric. The SATA Host controller (SATAHC) supports AHCI mode.

Figure 62 provides a block diagram illustrating the SATA Host Controller connectivity.

Figure 62: SATAHC Block Diagram



The SATA interface in this device is compatible with the SATA 6 Gbps standard. This section is based on the reference document SATA Revision 3.0 Specification. Refer to it for further details (https://www.sata-io.org/).

The SATA 3.0 Host controller registers are located in Appendix A.14, SATA 3.0 Host Controller Registers, on page 1027.

Copyright © 2015 Marvell

19.1 Features

This interface integrates the following features:

- Serial ATA 3.0 compliant
  - Gen3 Serial ATA PHY (6 Gbps) with speed negotiation to Gen1, Gen2
  - Supports Gen1i, Gen1m, Gen1x, Gen2i, Gen2m, Gen2x, Gen3
- Supports protocols over SATA ports
  - ATA and ATAPI commands
  - Native Command Queuing (NCQ)
    - In-order data delivery
    - 32 outstanding commands per port
  - Vendor Unique commands
- Standard AHCI interface with InBox AHCI driver

19.2 High-Level Functional Description—SATA Port Access

This subsection describes the methods of accessing SATA ports.

19.2.1 Detecting Device(s) and Obtain the Device Signature

If the Cold Port Detect interrupt has been enable, when a device has been detected trying to connect with the port, that port will generate an interrupt. If the interrupt is not enabled, the firmware /software should poll the status register of the port.

When SATA port status is 7Fh, the PHY is not ready yet. After the PHY is ready and before getting the signature FIS from the device, the status changes to 80h, after getting the signature FIS from the device:

- The status will be updated to the status content in the signature FIS
- The BSY bit is cleared
- The DRDY bit is set.

After the status changes to 50h/40h or the port receives the Port Detect interrupt, the software must read the signature register of that port and classify the device.

19.2.2 Issuing Commands to the Host Controller

After the signature has been read, the software should start to issue commands to the controller. To start the control just set the Start bit of the command register for the specific port.

The software should set the basic command and data related registers of the controller, like the Command List Base Address and FIS Base Address. See:

- Command List Base Address Lower 32-bits Register (i=0–1, n=0–1) (Table 1272 p. 1037)
- Command List Base Address Upper 32-bits Register (i=0–1, n=0–1) (Table 1275 p. 1039)
- Received FIS Base Address Lower 32-bits Register (i=0–1, n=0–1) (Table 1277 p. 1040)
- Command List Base Address Upper 32-bits Register (i=0–1, n=0–1) (Table 1275 p. 1039)

The software should:

1. Construct the command list header in the proper location relative to the Command List Base Address.
2. Fill with proper contents of the command list header.

Copyright © 2015 Marvell

3. Construct the command FIS and ATAPI command for the ATAPI device and the Physical Region Descriptor (PRD) table for data transfer command.
4. If it is a write command, the software should prepare the data in the memory related to the data base address.

When the command preparation is completed, the software can set the command issue bit corresponding to the slot of the command. Therefore, if the command header has been constructed in slot 5, the bit[5] should be set to 1 in the command issued register. The slot number of the command header is the bit position of the command issued register, that is they are in one-to-one mapping.

- If the issued command is First Party DMA command, the software should set the SATA Active (SActive/SCR3) Register (i=0–1, n=0–1) (Table 1288 p. 1052) in the same bit position as the Command Issue Register (i=0–1, n=0–1) (Table 1289 p. 1052).
- If there is any bit set in the Command Issue Register (i=0–1, n=0–1) and the START bit is also set, the host controller will start to load the command header and command FIS from the memory and send the command FIS to the device. If it is an ATAPI command, the controller will load the ATAPI command from the memory and send it to the device.
- If there is data transfer in the command, the controller will access the memory either to read the data from the memory (for a data write command) or put the data into the memory (for a data read command).
- After the command is finished, if the interrupt is enabled, there will be an interrupt pending. For Native Command Queuing (NCQ), that is, First Party DMA, one interrupt is not necessarily just related to one command-done event, the software should the check the SATA Active (SActive/SCR3) Register (i=0–1, n=0–1) to determine how many commands are complete.

19.2.3 Enumerating Port Multiplier Device and Attached Device(s)

Software should check if there is a port multiplier connecting to the SATA port after the device is detected. The software should perform the following procedure.

1. Send a SRST to the device with port number F.
2. Poll the status register until it is 50h/40h and read the signature register to check if the device is a Port Multiplier.
3. Use the Read Port Multiplier command to read the GSCR[2] register of the Port multiplier.
4. Use the Write Port Multiplier command to write PSCR[2] to enable all the ports that the multiplier supports.
5. Use the Read Port Multiplier command to read the PSCR[0] to check if the there is any device connecting to the port.
6. If there is device connect to the port, use the Write Port Multiplier command to clear the X bit in PSCR[1]. The X bit indicates the device presence has changed.
7. Software can start to access the device with the specific Port Multiplier port number as usual for the device.

19.2.4 Software Considerations

After software correctly prepares the command and the data, and sets the <Cmd Issue> field in the Command Issue Register (i=0–1, n=0–1) (Table 1289 p. 1052), the software awaits for the command-done interrupt. The host controller takes care of the command sending and command execution. There are some differences in processing for different types of commands.

Non-data ATA Commands

The software prepares the command header and the command FIS. There is no PRD table. To issue the command, the software sets the corresponding bit of the Command Issue Register (i=0–1, n=0–1). The software can read the status after the command is completed and take proper action if there is an error.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

PIO/DMA ATA DATA Commands

If the command is a write command, the software should prepare the:

- Command header
- Command FIS
- PRD table
- DATA in the memory

To issue the command, it sets the corresponding bit of the Command Issue Register (i=0–1, n=0–1). The software should read the status register after the command is completed and take proper action if there is an error.

Non-data ATAPI Commands

The software prepares the:

- Command header
- Command FIS
- ATAPI command (CDB)

There is no PRD table. To issue the command, it sets the corresponding bit of the Command Issue Register (i=0–1, n=0–1). The software can read the status after the command is completed and take proper action if there is an error.

PIO/DMA ATAPI Commands

If the command is a write command, the software should prepare the:

- Command header
- Command FIS
- ATAPI command (CDB)
- PRD table
- DATA in the memory

To issue the command, set the corresponding bit of the Command Issue Register (i=0–1, n=0–1). The software should read the status register after the command is completed and take proper action if there is error.

Native Command Queuing (NCQ) Commands

If the command is a write command, the software should prepare the:

- Command header
- Command FIS
- ATAPI command (CDB)
- PRD table
- DATA in the memory

To issue the command, it should set the corresponding bit of the Command Issue Register (i=0–1, n=0–1) and SATA Active (SActive/SCR3) Register (i=0–1, n=0–1). The software should read the status register after the command is completed and take proper action if there is an error.

19.2.5 Command Issued (CI) and SATA Active Registers

The function of these 2 registers is:

- The Command Issue Register (i=0–1, n=0–1) is the trigger and record for the host controller.
- The SATA Active (SActive/SCR3) Register (i=0–1, n=0–1) is the record of the NCQ command for the software.

The software can issue:

- 32 commands for one set of CI and SACT registers
- Up to 128 commands for four sets these registers.

Non-NCQ commands

For non-NCQ commands, the command will be sent and executed one by one for a certain device. If there is a port multiplier, each of the devices can have one outstanding command. The devices execute the command independently, but determination of which device transfers the data depends upon which port wins the arbiter of the port multiplier.

- The <Cmd Issue> clears after the command is complete.
- The software can issue a new command whenever this is bit is empty in the Command Issue Register (i=0–1, n=0–1).

NCQ commands

For NCQ command, each device can have up to 32 commands outstanding. For each command, the data can be partially transferred.

- For NCQ commands, the <Cmd Issue> clears after the NCQ command has been sent to the device and the host controller gets a REGD2H with BSY=0 and DRQ=0 from the device.
- For NCQ commands, the SATA Active (SActive/SCR3) Register (i=0–1, n=0–1) bit will be clear if the host received and Set Device Bit FIS with the corresponding bit is set to 1.
- For NCQ commands, the software can issue a new command when the bit is empty in both Command Issue Register (i=0–1, n=0–1) and SATA Active (SActive/SCR3) Register (i=0–1, n=0–1).

19.3 AHCI Mode

All of the AHCI mode interface registers are located in the SATA controller.

19.3.1 Operating Sequence

The basic AHCI mode operating sequence is as follows:

1. The host driver:
   - Programs all the registers pointing to the command and data structure.
   - Prepares the command and data for the SATA ports.
2. The host programs Command Issue Register (i=0–1, n=0–1) (Table 1289 p. 1052) to issue the command.
   The SATA controller reads in the data-memory structure when commands are received.
3. The SATA controller sends the command to the device and automatically performs the data transfer.
4. The SATA controller raises the interrupt to the host driver, and the host driver checks the status. After the status check, the command is complete.

Document Classification: Proprietary Information

19.3.2 AHCI Data-Memory Structure

The AHCI data-memory structure consists of the:

- SATA Received FIS
- SATA Command List
- SATA Command Table

19.3.2.1 SATA Received FIS Structure

The following registers specify table location:

- Received FIS Base Address Lower 32-bits Register (i=0–1, n=0–1) (Table 1277 p. 1040)
- Received FIS Base Address Upper 32-bits Register (i=0–1, n=0–1) (Table 1278 p. 1040)

Table 56 describes the SATA received FIS structure.

Table 56: SATA Received FIS Structure

Offset	Contents
00h	DMA Setup FIS
1Ch	Reserved
20h	PIO Setup FIS
34h	BIST Activate FIS
40h	Register D2H FIS
54h	Reserved
58h	Set Device Bit FIS
60h	Unknown FIS (up to 64 bytes)
A0h–FFh	Reserved

19.3.2.2 SATA Command List Structure

The following registers specify table location:

- Command List Base Address Lower 32-bits Register (i=0–1, n=0–1) (Table 1272 p. 1037)
- Command List Base Address Upper 32-bits Register (i=0–1, n=0–1) (Table 1275 p. 1039)

Table 57 describes the SATA command list structure.

Table 57: SATA Command List Structure

Offset	Contents
000h	Command 0 Header
010h	Reserved
020h	Command 1 Header
030h	Reserved
⋮	⋮
3C0h	Command 30 Header
3D0h	Reserved
3E0h	Command 31 Header
3F0h	Reserved

Copyright © 2015 Marvell

Table 58 describes the SATA command header structure for DW0 through DW7.

Table 58: SATA Command Header Structure

Dword (DW)	Byte 3	Byte 2	Byte 1	Byte 0
DW0	PRD Entry Number		PMP	CTRL	CFL - CMD FIS RD Length
DW1	PRD Byte Count			
DW2	Command table base address			Intf sel
DW3	Command table base address upper 32 bits			
DW4–DW7	Reserved			

Each DMA Command Header, DW0 through DW3, is outlined in Table 59 to Table 62.

Table 59: SATA Command Header DW0

Bits	Field Name	Description
31:16	PRD_ENT_NUM	Physical Region Descriptor Table Length This is the length of the scatter/gather descriptor table in entries, called the Physical Region Descriptor Table. Each entry is 4 Dwords. A value of 0h represents 0 entries; FFFFh represents 65,535 entries. The HBA uses this field to determine when to stop fetching PRDs. If this field is 0h, no data transfer occurs with the command.
15:12	PORT_MULT_PORT	Port Multiplier Port These bits indicate the port number that should be used when constructing each data FIS on transmit, and to check against each FIS received for this command.
11	RSVD	Reserved
10	CLEAR_BUSY	Clear Busy upon R_OK (C) When set, the HBA clears PxTFD.STS.BSY and PxCI.CI(pIssueSlot) after transmitting this FIS and receiving R_OK. When cleared, the HBA does not clear PxTFD.STS.BSY or PxCI.CI(pIssueSlot) after transmitting this FIS and receiving R_OK.
9	BIST	BIST (B) When 1, indicates that the command that software built is for sending a BIST FIS. The HBA sends the FIS and enters a test mode. The tests that can be run in this mode are outside the scope of this specification.
8	RESET	Reset A value of 1h indicates that the command from the software is for device reset. If necessary, the HBA performs a sync escape to put the device into an idle state before sending the command.

Document Classification: Proprietary Information

Table 59: SATA Command Header DW0 (Continued)

Bits	Field Name	Description
7	PREFETCH	Prefetchable (P) This bit is only valid if the PRDTL field is non-zero or the ATAPI 'A' bit is set in the command header. When set and PRDTL is non-zero, the HBA may prefetch PRDs in anticipation of performing a data transfer. When set and the ATAPI 'A' bit is set in the command header, the HBA may prefetch the ATAPI command. System software does not set this bit when using native command queuing commands or when using FIS-based switching with a Port Multiplier. **NOTE:** The HBA may prefetch the ATAPI command, PRD entries, and data regardless of the state of this bit. However, it is recommended that the HBA use this information from software to avoid prefetching needlessly.
6	WRITE	Write (W) When set, indicates that the direction is a device write (data from system memory to device). When cleared, indicates that the direction is a device read (data from device to system memory). If this bit is set and the P bit is set, the HBA may prefetch data in anticipation of receiving a DMA Setup FIS, a DMA Activate FIS, or PIO Setup FIS, in addition to prefetching PRDs.
5	ATAPI	ATAPI When 1, indicates that a PIO setup FIS will be sent by the device, indicating a transfer for the ATAPI command.
4:0	CMD_FIS_LEN	Command FIS Length (CFL) These bits indicate the length of the Command FIS. A value of 0h represents 0 Dwords, 4h represents4 Dwords. A length of 0 or 1 is illegal. The maximum allowed value is 10h, or 16 Dwords. The HBA uses this field to determine the length of the FIS to send to the device.

Table 60: SATA Command Header DW1

Bits	Field Name	Description
31:0	PRDBC	Physical Region Descriptor Byte Count (PRDBC) Indicates the current byte count that transfers on device writes (system memory to device) or device reads (device to system memory).

Table 61: SATA Command Header DW2

Bits	Field Name	Description
31:2	CMD_TBL_BASE_ADDR_L[31:2]	Command table descriptor base address lower 32 bits. **NOTE:** This address must be aligned to an 128-byte cache line, indicated by bits 6:0 being reserved.
1	RSVD	Reserved

Copyright © 2015 Marvell

Table 61: SATA Command Header DW2 (Continued)

Bits	Field Name	Description
0	RSVD	Reserved

Table 62: SATA Command Header DW3

Bits	Field Name	Description
31:0	CMD_TBL_BASE_ADDR_H[63:32]	Command table descriptor base address upper 32 bits.

19.3.2.3 SATA Command Table

The SATA command table is stored in memory locations offset from the SATA command table base address. Table 63 describes the SATA Command Table.

Table 63: SATA Command Table Content

Offset	Contents
00h	Command FIS (up to 64-bytes)
60h	Reserved
80h	Physical Region Descriptor Table (up to 65,536 entries).

19.3.2.4 Physical Region Descriptor (PRD)

Table 64 shows the PRD table command structure that stores information for each individual command.

Table 64: Physical Region Descriptor Table Command Structure

Dword (DW)	Bit	Byte 3	Byte 2	Byte 1	Byte 0
Item 0	0	Data Base Address			
	1	Data Base Address Upper 32 bits			
	2	Reserved			
	3	Reserved		Transfer Byte Count[21:0][1]	
Item 1	4	Data Base Address			
	6	Data Base Address Upper 32 bits			
	7	Reserved			
	7	Reserved		Transfer Byte Count[21:0][1]	
(Up to item 65,535)					

1. The actual Transfer Byte Count (TBC) = TBC field +1, for example when 511 is written, really 512b will transfer.

Document Classification: Proprietary Information

Table 65 describes the Physical Region Descriptor table for DW0–DW3.

Table 65: Physical Region Descriptor—DW0–DW3

Dword (DW)	Description
DW0	Data Base Address Lower 32 bits.
DW1	Data Base Address Upper 32 bits.
DW2	Reserved
DW3	[31]: Interrupt on completion of the PRD table. [30:22]: Reserved [21:0]: Transfer byte count.

19.4 Configuring Amplitude and Pre-emphasize Level

To select the amplitude/pre-emphasize AMP/EMPH/Slew level:

- In SATAGen1 mode, configure the Generation 1 Setting 0 Register (n=0–6) (Table 1494 p. 1216).
- In SATAGen2 mode, configure the Generation 2 Setting 0 Register (n=0–6) (Table 1496 p. 1218).
- In SATAGen3 mode, configure the Generation 3 Setting 0 Register (n=0–6) (Table 1498 p. 1220).

Copyright © 2015 Marvell

20 Audio (I^2S / S/PDIF) Interface

The device integrates an Audio Unit (ADU) that supports recording and playback for the following audio protocols:

- Integrated Interchip Sound (I^2S) with 3 different protocols:
  - Plain I^2S is the original Philips protocol
  - Right justified
  - Left justified
- Sony/Philips Digital Interconnect Format (S/PDIF) Consumer Mode IEC 60958-3 (Playback only)
- IEC 61937 Non-PCM protocols above the S/PDIF layer

Record: The ADU can record audio using the I^2S input protocol.

Play: It can play audio using either the I^2S or S/PDIF output protocols, or using both protocols simultaneously.

By using 2 separate DMA engines, the ADU supports concurrent recording and playback.

To select the audio mode (I^2S or S/PDIF), configure the <AudioMode> field in the SoC Control 1 Register (Table 1888 p. 1395).

Figure 63 provides a block diagram of the Audio unit.

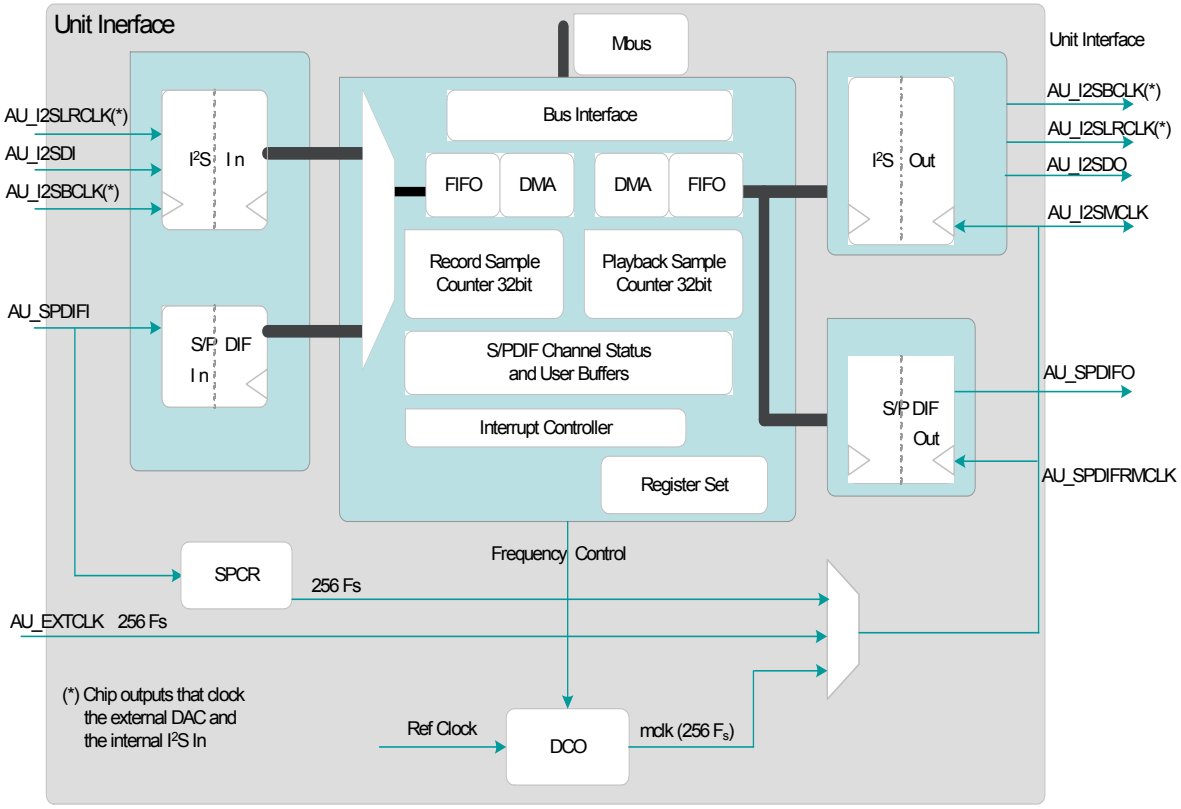


Note

For this device, recording is only supported by the I2S protocol.
Therefore, in Figure 63, ignore the S/PDIF input arrow and the S/PDIF control block.

Document Classification: Proprietary Information

Figure 63: Audio Unit Block Diagram



The Audio Unit registers are located in Appendix A.15, Audio (I2S / S/PDIF) Registers, on page 1054.

20.1 Features

The Audio Unit provides the following features:

- If Pulse Code Modulation (PCM) data is being played, the I2S and S/PDIF outputs can be active simultaneously. In this scenario, the same audio data is passed to the I2S and S/PDIF outputs. If non-PCM data is being played over the S/PDIF output, the I2S output must be muted or disabled.
- The Digital Controlled Oscillator (DCO) is used for generating the I2S playback and recording clock. It can also be used for the S/PDIF playback clock. The DCO takes its reference clock from the internal core clock. It can be configured to work at each of the supported frequencies with a configurable runtime offset of ppm[1] resolution adjustments.
- I2S or S/PDIF outputs can work only with the programmable DCO.
- I2S or S/PDIF outputs can work with the SPCR, an external clock, or the programmable DCO.
- The ADU supports stereo and mono playback and recording. During mono recording, only one of the channels is recorded. During mono playback, both channels may be used to concurrently play the same data, or the either of the channels can be muted resulting in the data playing in the single un-muted channel.
- Recording and playback may also be muted. Playback mute is possible on each of the interfaces separately.

1. ppm = Parts Per Million

- The ADU utilizes 2 DMA engines:
  - One recording DMA that is used to transfer data from the I^2S interface to memory.
  - One playback DMA that is used to transfer data from memory to either the I^2S interface, the S/PDIF interface, or both interfaces.
- The DMA engines operate at the core clock speed, and can be configured to have a burst transfer of 32 bytes or 128 bytes during memory access.
- When recording or playback is activated, the DMA engines work independently using the configured start address and the size of a cyclic buffer in memory.
- The DMA operation continues without the need for software intervention until playback or recording is stopped or paused. However, software intervention is required for moving data from the recording cyclic buffer to another memory location during recording, and to write new data in the playback cyclic buffer during playback.

I^2S Supported Features

For I^2S protocol support, the unit supports the following features:

- An audio sample rate (F_s) of 44.1/48/96 kHz.
- The I^2S input and I^2S output can only operate at the same sample rate.
- The available sample sizes are 16-bit, 20-bit, 24-bit, and 32-bit.
- Regardless of the sample size, 32-bits are always used per channel.
- Two channels are supported.
- Sample sizes do not have to be the same for input and output. Unused data bits are padded with zeros. For example, 8-bit padding is used for 24-bit sample sizes and 16-bit padding for 16-bit sample sizes.
- For the receiver and the transmitter side, the ADU outputs AU_I2SMCLK (master clock = $256xF_s$) and functions as a master over ADU_I2SLRCLK (1xFs) and over AU_I2SBCLK ($64xF_s$).
- Works in plain I^2S, right justified, and left justified formats.

S/PDIF Supported Features

For S/PDIF protocol support, the unit supports the following features:

- The IEC 60958-1, IEC 60958-3, and IEC 61937 specifications.
- An audio sample rate (F_s) of 44.1/48/96 kHz.
- Sample sizes of 16-bit, 20-bit, and 24-bit
- Outputs S/PDIFRMCLK = Recovered S/PDIF master clock (master clock = $256xF_s$)
- Double buffering for status and user bits at playback.
  - For playback, an interrupt can be generated when a new block playback begins. This allows the user approximately a one block playback period to configure the new channel status and user bits of the next block before these bits are sent.
  - All interrupts can be masked. S/PDIF control bit related interrupts can be masked per channel.

Copyright © 2015 Marvell

20.2 Recording Data Flow

To record data, follow these steps:

1. Configure the <Recording Buffer Start Address> field in the Recording Start Address Register (Table 1321 p. 1070) and the <Recording Buffer Size> field in the Recording Buffer Size Register (Table 1322 p. 1071).
2. Configure the:
   - <Recording Base> field in the Recording Window Base Address Register (Table 1293 p. 1055)
   - Recording Window Control Register (Table 1294 p. 1055)
   - <Recording DMA Burst Size> field in the Recording Control Register (Table 1320 p. 1069) and the required audio configuration modes using:
     - the Recording Control Register (Table 1320 p. 1068)
       and
     - the I2S Recording Control Register (Table 1324 p. 1071).
   - Configure the I^2S clock source by selecting one of the following clock sources through the <MCLK source> field in the Clocks Control Register (Table 1300 p. 1058):
   - Configurable DCO clock: The dco_lock should be read high before playback is enabled.
   - External clock



Note

- Each of these clocks has a frequency of $256xF_S$.
- When I^2S data is recorded, the ADU generates the I^2S clocks from the configured clock source.

3. Decide which buffer management method to use. Enable the relevant mask bits and optionally set a proper value in the Recorded Byte Counter for Interrupt Register (Table 1309 p. 1062). The recording buffer management methods are described in section Section 20.2.1, Recorded Data Buffer Management Methods, on page 254.
4. Enable recording by asserting the <I2S Recording Enable> field in the Recording Control Register (Table 1320 p. 1068).

Based on the protocol being used, the DMA transfers data to the cyclic memory buffer when the I^2S interface is enabled.

The size of each DMA write burst access is determined by the recording DMA burst size configuration.

20.2.1 Recorded Data Buffer Management Methods

There are two methods used by the software to move recorded data from the cyclic buffer to other memory locations before the recording DMA overwrites the buffer.

Divided Buffer Method

As shown in Figure 64, the buffer is divided into quarters or halves. After each quarter or half of the buffer is written by the recording DMA, the software is interrupted. An interrupt handler can then move the respective buffer segment to another memory location before it is overwritten, when the DMA address counter wraps around.

In this method, if an interrupt handler operation is delayed for a period of less than a full buffer recording time, the cause register still holds information of which quarters of the buffer were recorded. This provides the software an opportunity to recover from an unexpected interrupt handler delay and prevents data being dropped.

Configured Recorded Byte Counter Method

It is possible to configure the recorded byte counter for an interrupt.

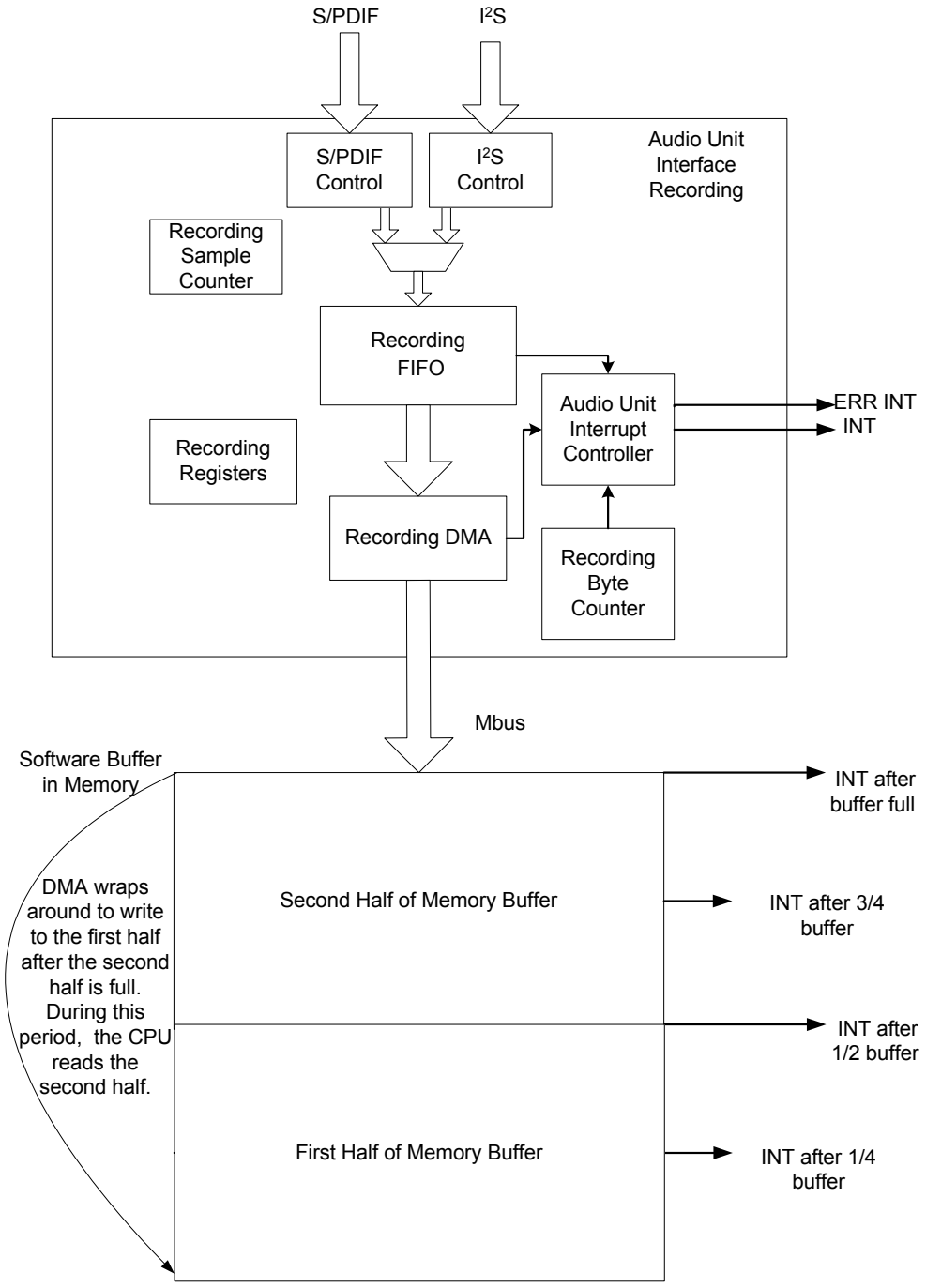
Each time the dedicated counter reaches a pre-configured value in the Recorded Byte Counter for Interrupt Register (Table 1309 p. 1062), an interrupt is asserted in the Audio Interrupt Cause Register (Table 1304 p. 1059), and the counter is reset. This allows for user-defined interrupt resolutions.

The interrupt cause bits can be read from the Audio Interrupt Cause Register (Table 1304 p. 1059). The respective mask bits can be found at the Audio Interrupt Mask Register (Table 1305 p. 1061).

For this device, recording is only supported on the I^2S protocol. Therefore, in Figure 64, ignore the S/PDIF input arrow and the S/PDIF Control block.

Copyright © 2015 Marvell

Figure 64: Recording Flow



20.2.2 Recovering Residual Data

After recording data is disabled, regardless of which buffer management method is used, residual recorded data remains in the memory buffer. This residual data enters the memory in the period between the assertion of a recording data interrupt and the disabling of the interface.

With the divided buffer method, it is possible to determine how many recorded bytes remain by reading the <Recording Buf Byte Count> field in the Recording Buffer Byte Counter Register (Table 1323 p. 1071). This field holds the number of bytes written by the recording DMA, since the last recording byte counter wrap around.

With the byte counter method, the <Recorded Byte Count for Interrupt> field in the Recorded Byte Count for Interrupt Register (Table 1306 p. 1061) can be used for the same purpose.

20.2.3 I^2S Recording

The ADU is always master over the I^2S clocks. When I^2S recording is enabled, the ADU activates the I^2S clocks for both the external transmitter and the ADU recording sub-unit.

20.2.4 Recording Mute

When recording mute is enabled by the <Recording Mute> field in the Recording Control Register (Table 1320 p. 1069), the I^2S recorded data value is replaced by zeros.

The recording DMA and the I^2S interface continue operating normally.

20.2.5 Recording Pause

When recording pause is enabled in the <Recording Pause> field in the Recording Control Register (Table 1320 p. 1069), the I^2S interface continues to work normally but the recording DMA halts. Unlike the I^2S recording disable (STOP), the I^2S clocks remains active as long as I^2S recording is enabled. This allows the I^2S transmitter to continue playing data even though the ADU discards it.

Note

If I^2S recording is disabled by setting the <I2S Recording Enable> field in the Recording Control Register Recording Control Register (Table 428 p. 386) (Table 428 p. 389) to 0, the I^2S clocks remains active if the I^2S playback is enabled, because the I^2S master mode is used.

20.3 Playback Flow

To playback data, use the following procedure:

1. Select one of the following clock sources through the <MCLK source> field in the Clocks Control Register (Table 1300 p. 1058). Each of these clocks has the frequency of $256xF_S$:
   - Configurable DCO clock.
   - External clock
2. Configure the <Playback Buffer Start Address> field in the Playback Start Address Register (Table 1311 p. 1065) and the <Playback Buffer Size> field in the Playback Buffer Size Register (Table 1312 p. 1065).
3. Write the playback data to the entire cyclic buffer.
4. Configure the following:
   - <Playback Base> field in the Playback Window Base Address Register (Table 1295 p. 1056)
   - Playback Window Control Register (Table 1296 p. 1056)
   - <Playback DMA Burst Size> field in the Playback Control Register (Table 1310 p. 1062)
   - Required audio configuration modes in the:
     – Playback Control Register (Table 1310 p. 1062)

Copyright © 2015 Marvell

  – S/PDIF Playback Registers (see Table 1314 through Table 1318)
  – I2S Playback Control Register (Table 1319 p. 1068)

5. Decide which buffer management method to use. Enable the relevant mask bits and optionally set a proper value in the Playback Byte Counter for Interrupt Register (Table 1308 p. 1061). The playback buffer management methods are described in section Section 20.3.2, Playback Data Buffer Management Methods, on page 258.
6. Enable playback with either the <I2S Playback Enable> or the <SPDIF Playback Enable> field in the Playback Control Register (Table 1310 p. 1064).

It is possible to enable the I^2S interface, the S/PDIF interface, or both interfaces simultaneously. If both interfaces are enabled, they should be enabled and disabled at the same time.

During simultaneous playback, the DMA sends the same data to both I^2S and S/PDIF interfaces. S/PDIF control bits are sent only to the S/PDIF interface when a playback sample size of 24-bits or less is selected. Non-PCM data cannot be used during simultaneous playback.



Note

When playing a 32-bit sample, disable or mute the S/PDIF output interface.

As long as playback is activated, playback-DMA transfers data independently from the cyclic memory buffer to the ADU outputs. The size of each DMA read burst access is determined by the <Playback DMA Burst Size> field in the Playback Control Register (Table 1310 p. 1062).

20.3.1 PLL Configuration

The following procedure describes how to configure the PLL to get the correct behavior and desired MClk frequency.

1. Adunit Apll Frequency Offset Valid
   Reg => 0XD0018414
   Bits [16] => 0x0

2. Adunit Apll Sw Reset
   Reg => 0XD0018414
   Bits [31] => 0x0 (Enable)

3. Wait for at least 30 clock cycles.

4. For 44.1KHz sample rate -> 11.290MHz DCO frequency:
   Reg => 0XD0018410
   Bits [9:1] => 0x5
   Bits [10:18] => 0x1B
   Bits [22:19] => 0x5

   Reg => 0XD0018414
   Bits [15:0] => 0x735

   Reg => 0XD0018418
   Bits [6:0] => 0xC

5. For 96KHz sample rate -> 24.576MHz DCO frequency:
   Reg => 0XD0018410
   Bits [9:1] => 0x5

Copyright © 2015 Marvell

Bits [10:18] => 0x1D
Bits [22:19] => 0x5

Reg => 0XD0018414
Bits [15:0] => 0x22B0

Reg => 0XD0018418
Bits [6:0] => 0X6

- To get 12.288MHz (48KHz sample rate), use the same configuration as in 24.576MHz except for:
  Reg => 0XD0018418
  Bits [6:0] => 0Xc

- To get 49.152MHz, use the same configuration as in 24.576MHz except for:
  Reg => 0XD0018418
  Bits [6:0] => 0X3

20.3.2 Playback Data Buffer Management Methods

The software must continuously fill the cyclic memory buffer using one of the following interrupt methods. Interrupt cause bits can be found at the Audio Interrupt Cause Register (Table 1304 p. 1059) and can be masked by using the Audio Interrupt Mask Register (Table 1305 p. 1061).

Divided Buffer Method

As shown in Figure 65, the first method divides the buffer into quarters or halves. After each quarter or half of the buffer is read by the playback DMA, the software is interrupted. An interrupt handler can overwrite the already played buffer segment with new playback data, while playback continues from the following buffer segments.

In this method, if an interrupt handler operation is delayed for a period of less than a full buffer playback cycle, the cause register still holds the information of which quarters of the buffer were played. This gives the software more time to recover from an unexpected interrupt handler delay.

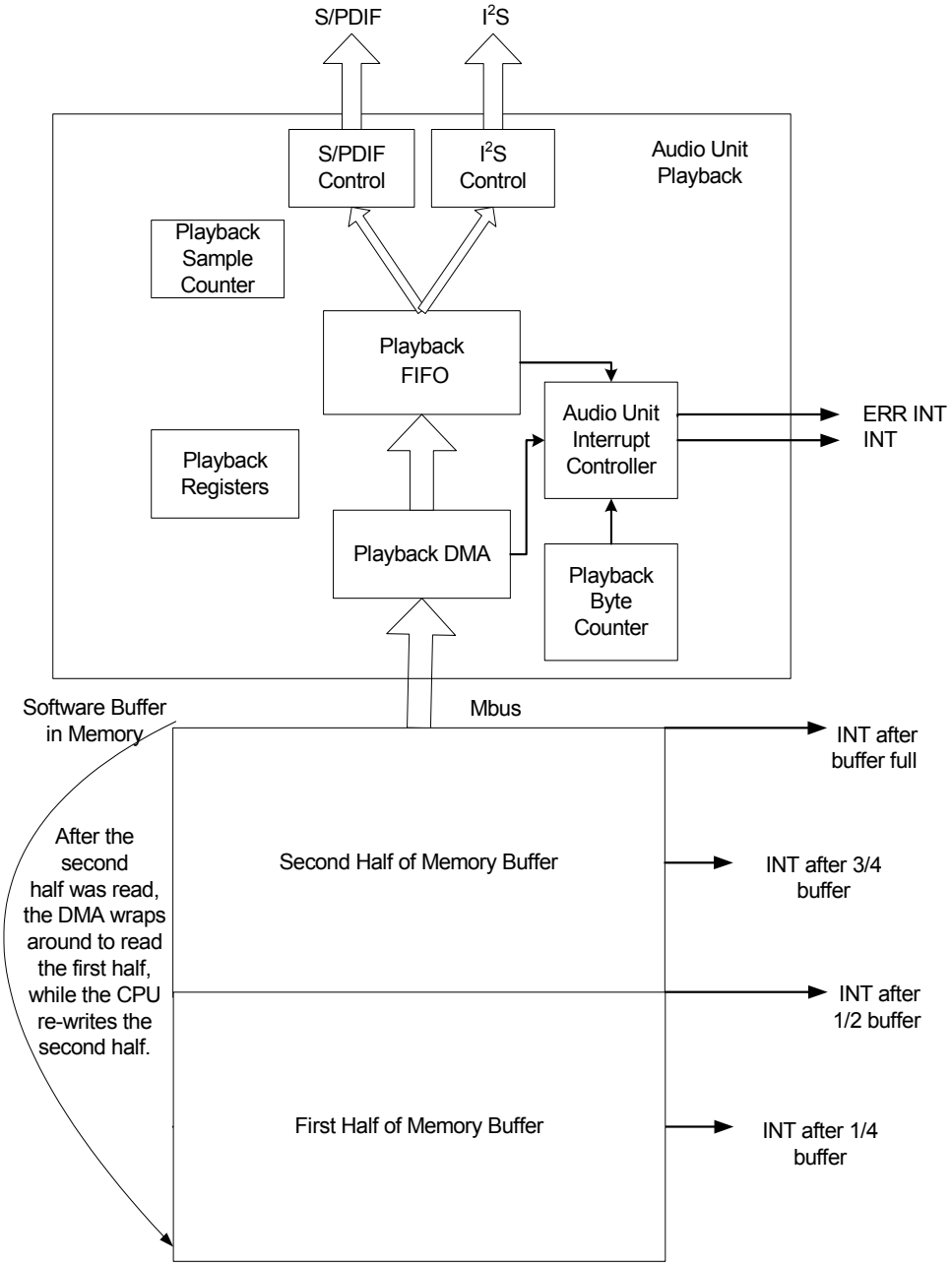
Configured Playback Byte Counter Method

The second method enables the user to configure the playback byte count for an interrupt.

Each time the dedicated counter reaches a pre-configured value in the Recorded Byte Counter for Interrupt Register (Table 1309 p. 1062), an interrupt is asserted in the Audio Interrupt Cause Register (Table 1304 p. 1059), and the counter is reset.
The advantage of this method is that it allows other user-defined interrupt resolutions.

Copyright © 2015 Marvell

Figure 65: Playback Flow



20.3.3 S/PDIF Specific Playback Flow

User bits can be sent either from the ADU internal buffers at:

- SPDIF Playback User Bits Left n Register (n=0–5) (Table 1317 p. 1067)
  and
- SPDIF Playback User Bits Right n Register (n=0–5) (Table 1318 p. 1067)

or from the memory. When status and user bits are sent from the ADU internal buffers, they are updated during playback in parallel to the data. Before the start of each new played block, an S/PDIF start of block interrupt can be enabled. The software uses this interrupt to update the new ADU internal buffers status and/or user bits. The new bits are played together with the block following the one that has now started.

Status and user bits are double buffered, allowing the software approximately one block of playback time to update the new S/PDIF user and/or status bits.

20.3.4 Playback Mute

It is possible to mute either the I^2S or the S/PDIF interface during simultaneous playback, or when only one of the interfaces is active.

During simultaneous playback with both interfaces enabled, the interfaces can be muted together or separately at any time. Muting is performed using the <I2S Playback Mute> field or the <SPDIF Playback Mute> field in the Playback Control Register (Table 1310 p. 1063).

During mute, the Playback-DMA and I^2S clocks remain active.

During non-simultaneous I^2S and S/PDIF playback control, when mute is activated, the S/PDIF user and status bits are automatically taken from the internal buffers. This means that the internal buffers should hold their required mute values before mute can be asserted.

20.3.5 Playback Pause

If the <Playback Pause> field in the Playback Control Register (Table 1310 p. 1063) is enabled, the playback DMA halts, but the I^2S and the S/PDIF interfaces remain active and playing “silence”.

During pause, the S/PDIF user and status bits are automatically played from the internal buffers. This means that the internal buffers should hold their required pause values before pause can be asserted.

20.3.6 Sample Counters and DCO Run-Time Operation

The ADU includes a <Playback Sample Counter> field in the Playback Sample Counter Register (Table 1298 p. 1057) and a <Recorded Sample Counter> field in the Recording Sample Counter Register (Table 1299 p. 1058).These counters are incremented by each incoming and outgoing audio sample from the ADU I/O interface.

Copyright © 2015 Marvell

20.4 Error Handling

The ADU error handling of recording parity errors, recording overrun, and playback underrun is described in the following sections.

Recording Overrun

When the internal recording FIFO is full, new samples are dropped. The <Recording Overrun Err> field in the Audio Error Cause Register (Table 1302 p. 1059) is asserted. This event may occur, for example, when the Audio unit write accesses to the DDR are slower than the recording speed.

The field <Recording Overrun Err> can be masked by de-asserting bit [4] in the Audio Error Mask Register (Table 1303 p. 1059).

Playback Underrun

If Playback underrun occurs, a Playback I^2S Underrun Error interrupt or a Playback S/PDIF Underrun Error interrupt in the Audio Error Cause Register (Table 1302 p. 1059) can be enabled via bit [5] and bit [6], respectively, in the Audio Error Mask Register (Table 1303 p. 1059).

20.5 Audio Unit Memory Structure

The memory structure for I^2S and S/PDIF data transmit and receive can use 16-, 20-, or 24-bit sample sizes. I^2S data transmit and receive may also use a 32-bit sample size.

For 16-bit playback and recording, 2 memory structure options exist:

16-Bit Compact Mode

Data is stored in memory sequentially and there are no gaps, see Example A (stereo) and Example C (mono) in Figure 66.

In this option, the 192 status and user bits per channel are double buffered in the ADU internal buffers. Validity, parity, and block start bits are not recorded.

On the playback path, the user and channel status bits are taken from the internal buffers that hold the 192 status bits and the 192 user bits per channel. These registers are configured per channel, and are sent automatically and repeatedly for every block.

For normal operations, the validity bit should be configured in the register file as:

- 0 for PCM traffic
- 1 for non-PCM traffic

Full 16-Bit Mode

This option is depicted in Example B (stereo) and Example D (mono) in Figure 66.

This option uses twice as much memory space and bandwidth versus the 16-bit Compact Mode. It allows the user to store status, validity, parity, block start and parity error indication bits values in memory, along with the data.

During playback the user bit, validity bit, and start of block can be played from memory on a sub-frame basis.



Note

Non-linear PCM traffic always uses an S/PDIF 16-bit sample size.

To determine the sample size mode, configure the <Playback Sample Size> field in the Playback Control Register (Table 1310 p. 1064) and the <Recording Sample Size> field in the Recording Control Register (Table 1320 p. 1070). For I^2S only, also configure the <I2S PB sample size> field in the I2S Playback Control Register (Table 1319 p. 1068), and the <I2S Recording Sample Size> field in the I2S Recording Control Register (Table 1324 p. 1071).

Figure 66 displays the memory structure for data transmit and receive in 16-/20-/24-bit sample size for I^2S and S/PDIF data.

Note

Before using the Audio unit to play a PCM file, the user must first ensure that the PCM file memory structure is compatible with the ADU memory structure. If not, the PCM file must be converted to the Audio unit memory structure.

For example, an audio .wav file may use the following memory structure per sample size:

- *16-bit:* Same as the Audio unit 16-bit Compact Mode.
- *20-bit:* Each 20-bit sample is stored in a 24-bit word, where bits[19:0] are stored in bits[23:4] of the 24-bits word. The four bits [3:0] are padded with zeros. The 24-bit word is stored in memory as described below.
- *24-bit:* Each of the 24-bits is stored sequentially in memory without padding.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Figure 66: Memory Structure for Transmit and Receive

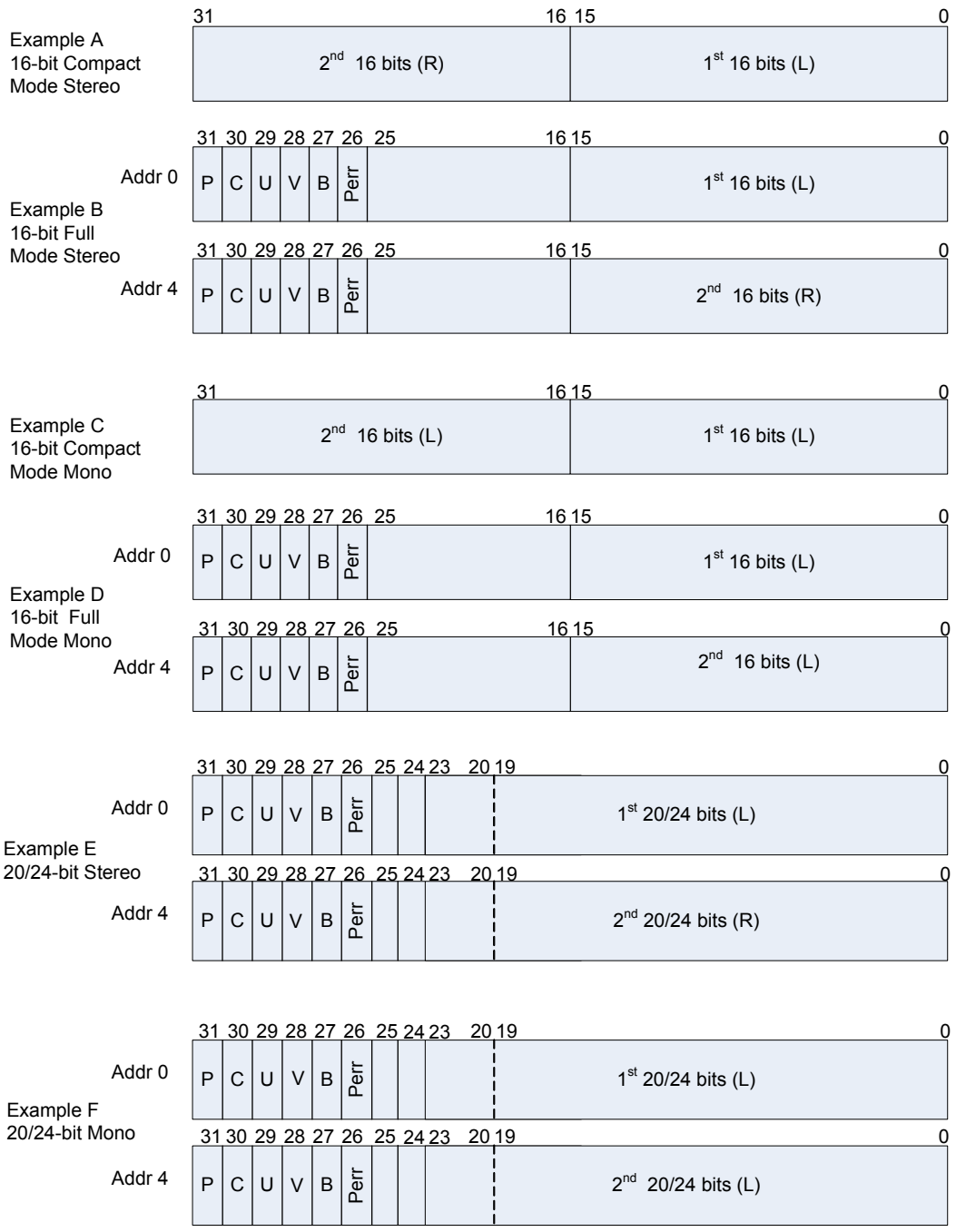


Table 66 describes the meaning of each memory structure bit for all sample sizes, except for 16-bit Compact mode.

Table 66: Audio Unit Memory Bit Description

Bit	Symbol	Name	Description
31	P	S/PDIF parity bit	This bit is recorded but not used during playback.
30	C	S/PDIF channel status	This bit is recorded but not used during playback. Channel status bits are used from the ADU internal buffers.
29	U	S/PDIF user bit	User bit is recorded and can be used during playback from the ADU internal buffers or from memory.
28	V	S/PDIF validity bit	When the validity bit is recorded or played from memory, it is always asserted by the ADU when the Perr bit is asserted, or in the case of recording/playback mute or pause. Validity can be used during playback from the ADU register file or from memory, as configured in the SPDIF Playback Control Register (Table 1314 p. 1066). When playing non-PCM audio, This bit should be asserted.
27	B	S/PDIF block start trigger	This bit is recorded and can be used during playback when the <SPDIF PB Block Start> field in the SPDIF Playback Control Register (Table 1314 p. 1066) is asserted. When asserted, the B-bit indicates to the ADU to send the data sample with the start of block preamble. Block start should be computed internally, and not taken from the B-bit during playback when both user and status bits are taken from the ADU internal buffers. Otherwise, the block start should be taken from memory. During underrun or pause, or during mute in non-simultaneous S/PDIF and I^2S playback, this bit is automatically computed by ADU and the B-bit is ignored.
26	Perr	S/PDIF parity error indication	Used in playback to assert validity in the event of a parity error indication. Perr can be used in playback to force a parity error in the outgoing sub-frame. If the <Force Parity Error> field in the SPDIF Playback Control Register (Table 1314 p. 1066) is asserted and the <Perr> bit is asserted, the ADU computes an erroneous parity. This can be used to maintain the parity error that was detected in the recording. **NOTE:** When Perr is asserted and PCM data was recorded, the recorded Perr bit may equal the calculated parity since the ADU asserts the recorded validity when a parity error is detected.

20.5.1 Stereo Mode Alignment

In Stereo mode recording and playback, the left channel data is stored in memory locations with addres2=0, and the right channel is stored in memory locations with address2=1.

When using the 16-bit Compact mode in Stereo mode, the left channel is stored in memory locations with address1=0, and the right channel is stored in memory locations with address1=1.

Copyright © 2015 Marvell

21 Serial Peripheral Interface (SPI)

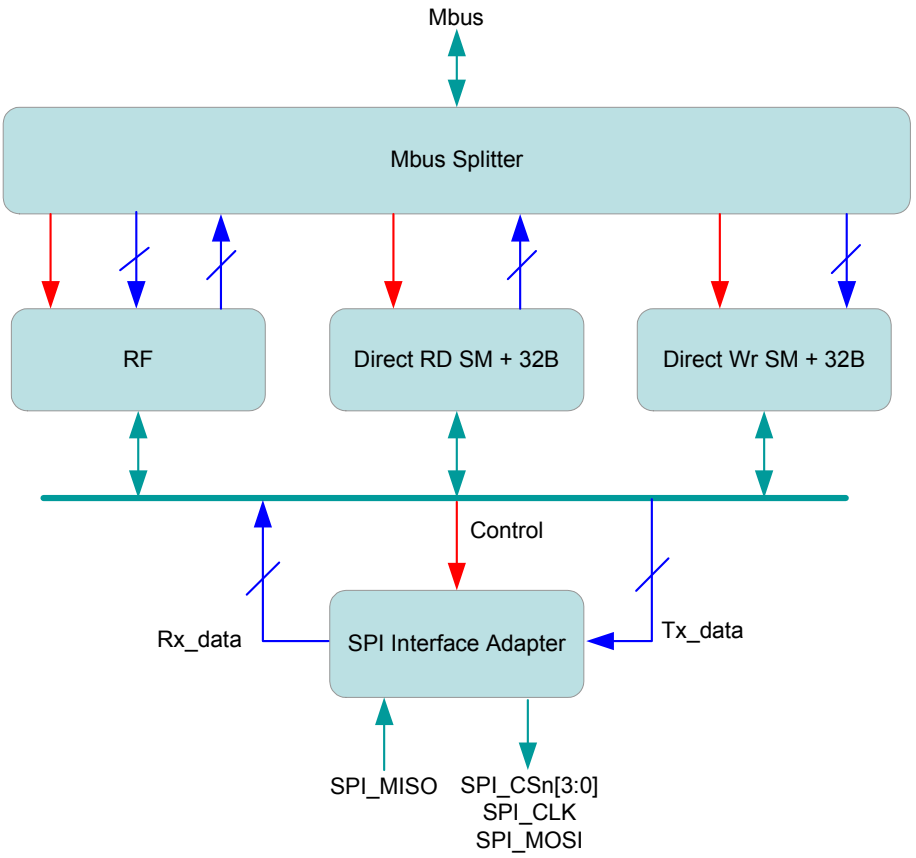
This section describes the general-purpose Serial Peripheral Interface (SPI) ports that are integrated into the device. The device integrates 2 SPI controllers, each with 4 chip selects multiplexed on the MPP pins.

The SPI is a synchronous serial data protocol used for transferring data simply and quickly from one device to another. With an SPI connection, there is always one Master device that controls the peripheral devices (Slaves). Data over the SPI bus is sent and received in parallel and is related to the internally generated baud rate clock.

The SPI registers are located in Appendix A.16, Serial Peripheral Interface (SPI) Registers, on page 1073.

Figure 67 shows a block diagram of the SPI controller.

Figure 67: SPI Controller Block Diagram



21.1 Features

The main features of the SPI include:

- Master-Slave protocol:
  - Source synchronous bus that allows the Master device to toggle the clock during transactions.
  - As data is being clocked out, new data is clocked in
  - The slaves are controlled by the master clock, and may not manipulate the clock
- Configurable clock rate with extra flexible speed points (down to as low as Core clock/1920)
- Configurable clock inactive state polarity (CPOL)
- Configurable data driving/sampling phase (CPHA)
- Supports direct access to SPI Slave:
  - CPU direct access to the memory space of the SPI
  - Exchange of up to 32B burst in a single CPU transaction.
  - Configurable Read opcode
  - Configurable Write opcode
  - Configurable address length (1–4 bytes)
  - Configurable presence of a dummy byte between address phase and data phase
- Supports Indirect access to SPI Slave: Exchange of 1 or 2B (read or write) using register access
- Chip Selects:
  - Support of up to 4 slave devices with dedicated chip selects
  - Boot considerations:
  - Supports boot 24 or 32-bit address
  - Default read opcode is 0x3
  - Default timing mode is 0 (CPOL = 0, CPHA = 0)
  - Default clock rate is Core clock/26

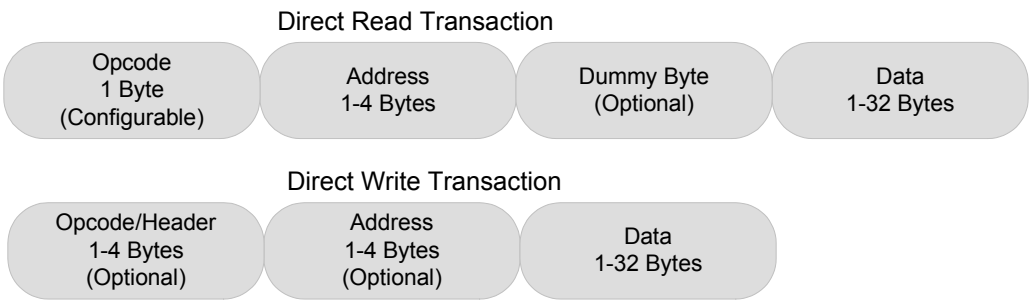
21.2 Functional Description

The SPI Slaves can be accessed directly or indirectly.

- Use the indirect access for most of the slave configurations
- Use the direct access to reduce software overhead for bulk read and write operations.

Figure 68 illustrates the direct access transaction level (read and write).

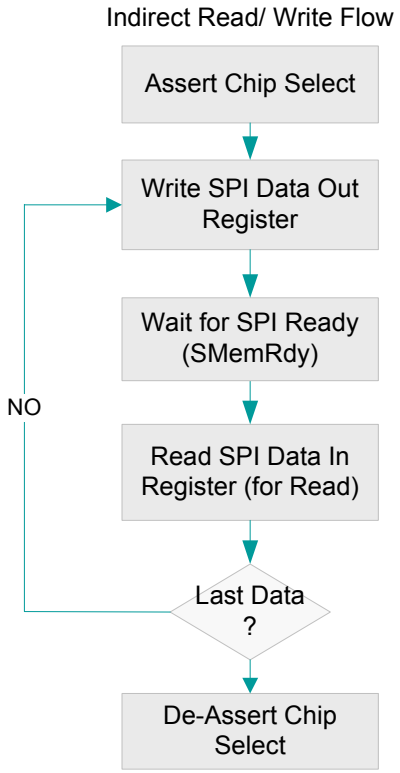
Figure 68: SPI Functional Diagram—Direct Access



Document Classification: Proprietary Information

Figure 69 illustrates indirect access transaction flow (read and write). It is possible for the software to control Chip-Select manually, or it can be asserted automatically for each transaction.

Figure 69: SPI Functional Diagram—Indirect Access



21.3 SPI Modes

Bus Signals

SPI signals are multiplexed on the Multi-Purpose Pins (MPPs). For full details, refer to the *Hardware Specifications* of the device.

Modes

The SPI supports 4 clock modes configurable by <CPOL> and <CPHA> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1074).

CPOL defines the clock inactive state (that is, if upon chip select (CS) assertion SPI_CLK is high or low).

CPHA selects between 2 options to transmit and sample data on SPI_CLK edges:

- CPHA = 0: Data is transmitted on even edges and sampled on odd edges, or
- CPHA = 1: Data is transmitted on odd edges and sampled on even edges.

Figure 70 shows the SPI timing in CPHA=0 for both CPOL=0 and CPOL=1.

Figure 70: SPI Timing for CPHA=0

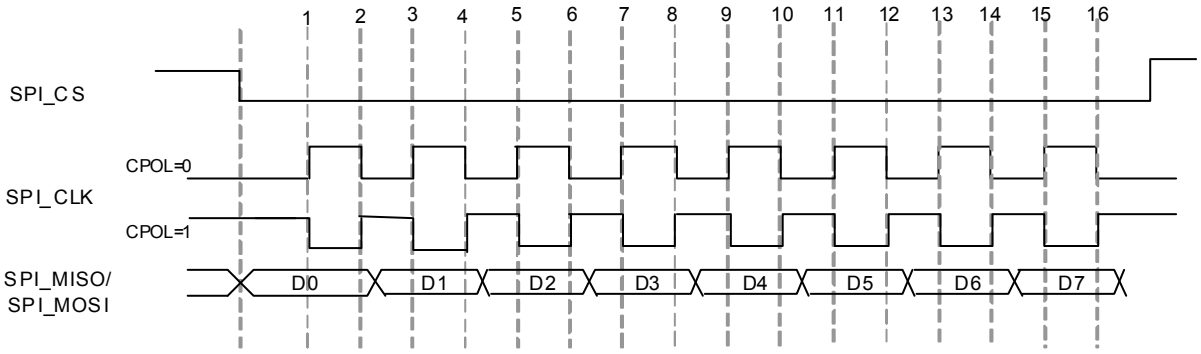
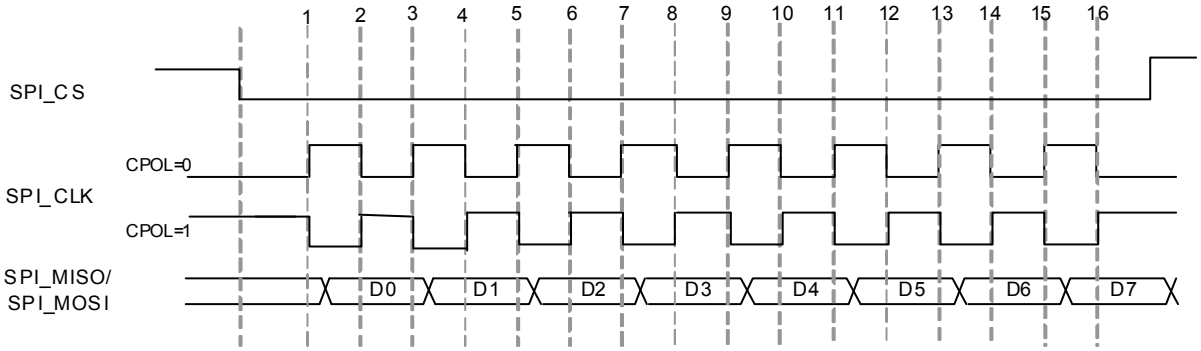


Figure 71 shows the SPI timing in CPHA=1 for both CPOL=0 and CPOL=1.

Figure 71: SPI Timing for CPHA=1



Copyright © 2015 Marvell

21.4 Indirect Access

This section describes the SPI Indirect access operations:

- SPI Input/Output
- Output 1 Byte to SPI
- Input 1 Byte from SPI
- Output or Input 2 Bytes

21.4.1 SPI Input/Output

The serial data clock is generated out of the device core clock (TCLK) with a programmable baud generator that has prescaler values of (2 ^ SPRR) * SPR, where SPRR<=7 and SPR = 2, 3, 4 … 15. At power up, the prescaler value is 26.

SPR is configured by setting the <SPI_SPR> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1075). SPRR is configured by setting the <SPI_SPPR0> and <SPI_SPPR_HI> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1075).

The following pseudo-code shows a simple SPI I/O setup:

```
for (spr = 2; spr <= 15; spr++) {
for (sppr = 0; sppr <= 7; sppr++) {
divider = spr * (1 << sppr);
```



Note

For the maximum allowable frequency, refer to the device Hardware Specifications.

The <SPI CS num> field in the SPI Control Register (N=0–1) (Table 1326 p. 1074) defines which SPI Chip-Select signal (SPI_CSn[N:0]) is asserted whenever the <CSnAct> field in the SPI Control Register (N=0–1) (Table 1326 p. 1074) is set. The Serial data clock output SPI_SCK toggles only during transactions.

21.4.2 Output 1 Byte to SPI

To output one byte, the CPU writes to the SPI Data Out Register (N=0–1) (Table 1328 p. 1076).

This register's bits [7:0] are shifted out on SPI<n>_MOSI on the transmitting edge as defined by the <CPOL> field and the <CPHA> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1074). There will be eight clock pulses output for eight data bits. The order of transmission is determined by <TxLSBF> in the same register. After all eight bits are shifted out, the <SMemRdy> field in the SPI Control Register (N=0–1) (Table 1326 p. 1074)is set. When the CPU again writes to the SPI Data Out Register (N=0–1), the Serial-Memory-Data-Ready status (<SMemRdy> field) is cleared.

21.4.3 Input 1 Byte from SPI

While SPI_SCK is toggling, the SPI<n>_MISO pin is sampled into the SPI Data In Register (N=0–1) (Table 1329 p. 1076).

Therefore, to read one byte from the external SPI device, the CPU can write a dummy data into the SPI Data Out Register (N=0–1). As dummy data is shifted out on SPI<n>_MOSI, data is shifted in from the SPI<n>_MISO pin. When the <SMemRdy> field is set, the CPU can read the SPI Data In Register (N=0–1) to retrieve the input data byte.

Copyright © 2015 Marvell

21.4.4 Output or Input 2 Bytes

To simplify firmware and improve the read/write throughput, the SPI interface logic can shift two bytes in and out for each access. When the <BYTE_LEN> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1075)is set to 0x1, two byte I/O mode is enable. When the CPU writes to the SPI Data Out Register (N=0–1), bits [7:0] are shifted out on SPI<n>_MOSI first, followed by bits [15:8].

As for data input, SPI<n>_MISO is shifted into the SPI Data In Register (N=0–1) [7:0] bits first, followed by bits [15:8].

When both bytes are shifted out (and in), the <SMemRdy> field is set.

21.5 Direct Mode

The device SPI controller supports direct read and writes from/to external SPI devices, without software overhead of reading and writing from/to the SPI Control Register (N=0–1) (Table 1326 p. 1074), SPI Data Out Register (N=0–1) (Table 1328 p. 1076), and SPI Data In Register (N=0–1) (Table 1329 p. 1076).

This section describes the 2 SPI Direct mode operations:

- Direct Read from SPI
- Direct Write to SPI

21.5.1 Direct Read from SPI

Two read modes are supported, as defined by the <DirectRdHighSpeed> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1075).

Read: Clear DirectRdHighSpeed: There is no dummy byte insertion after the address phase.

High Speed Read: Set DirectRdHighSpeed: (also known as fast-read) Dummy byte is inserted after the address phase.

The length of the address can be set to be between 1 and 4 bytes according to the <DirectAddrLen> field of that register.

If using SPI flash, set the <Attr> field in the CPU address decoding windows to match the SPI interface (port and CS number). Any CPU read to this address space is converted by the SPI controller to a SPI flash read transaction, composed of an address phase followed by a data phase.

The actual sequence that is driven on the SPI interface is:

1. Assert the SPI_CSn that corresponds to the <Attr> field.
2. Write command to the SPI.
   The 8 bit opcode is taken from the <DirectRdHeader>.
3. The Address is driven in 1–4 phases based on the configuration of the <DirectAddrLen> field.
   The order of the bytes is MSB to LSB.
4. In High Speed (HS) mode, as defined by <DirectRdHighSpeed>, add a 1B dummy write.
5. Read the data, according to the length given by the request.
6. De-assert the SPI_CSn signal.



Note

The maximum supported burst read is 32B.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

21.5.2 Direct Write to SPI

The device SPI Controller supports direct writes that will be transmitted on the SPI. The action and data to be transmitted consists of the following:

1. Assert CS.
   This stage can be omitted by clearing the <Direct Wr Deassert Cs> field in the SPI Direct Write Configuration Register (N=0–1) (Table 1334 p. 1078).
2. CS hold Gap.
   All signals are steady. The length of the gap will be at least <Direct Wr Cs Hold> core clock cycles.This gap will not exist when <Direct Wr Deassert Cs> is cleared.
3. Constant Header (optional):
   This is a 1-4 byte field that is taken from the <SpiDirectHdr> field in the SPI Direct Write Header Register (N=0–1) (Table 1335 p. 1079).
   The size of the header is configured on <Direct Wr Hdr Size> field in the SPI Direct Write Configuration Register (N=0–1) (Table 1334 p. 1078). The header can be omitted by clearing <Direct Write Hdr Enable> of that same register. A possible usage of this header is setting a Write command, that will be served as a prefix to any command.
4. Address Phase.
   This is a 1–4 byte field that is taken from the address of the request.
   The size of the address is configured via the <DirectAddrLen> field in the SPI Interface Configuration Register (N=0–1) (Table 1327 p. 1075).
   The entire Address phase is omitted when <Direct Wr Addr Enable> field in the SPI Direct Write Configuration Register (N=0–1) (Table 1334 p. 1078) is cleared.
5. Data Phase.
   A 1–32 byte transition of the data provided by the write request.
6. CS Preservation.
   The CS will be return to the state that it was prior to the Write command. If it was asserted before the action took place, then this stage is meaningless. If it was not asserted, then at this stage the CS will be de-asserted.
   This stage will be omitted if <Direct Wr Deassert Cs> is cleared.



Note

The maximum supported burst write is 32B.

21.5.3 SPI Timing

Separated Attributes per CS mode

The <SPI DecodeMode> field is 0x0.
While using this mode the address resolution and assignment to the appropriate CS is accomplished, according to the attribute settings defined in the corresponding address windows of the originating unit. To address two different CSs, the originating unit should have two different address decoding windows each referenced to the address space covered by the specific CS. For details of the decoding scheme, refer to:

- Table 3, CPU Interface Mbus Decoding Units IDs and Attributes, on page 46

Single Window mode The <SPI DecodeMode> field is 0x1.
In this mode, software defines a continuous address space that covers the entire physical space of the SPI devices. For example, if there are two SPI parts running over CS0 and CS1, each 128 MB in size, software will define the address space for these chip selects as address A to address A+256 MB. The software of the originating unit (for example, the CPU or PCIe) assigns a single address decoding window that covers address space A to A+256 MB (256 MB in size) without differentiating between the two SPI domains. This window can cover up to 8 CS.
Any transaction hits this SPI window (regardless of whether it originated in CS0 or CS1). Once it hits this window, it is routed internally to the SPI controller.
In the SPI controller, it is further decoded, and directed to the appropriate CS. This sub-decoding is accomplished according to 3 additional bits of the original address. For maximum flexibility, the location of these bits is configurable via the <SPI Window0IndexLSB> field in the SPI CS Address Decode Register (N=0–1) (Table 1337 p. 1080).

Double Window mode The <SPI DecodeMode> field is 0x2.
This mode uses the same method as the Single Window mode, but provides further address decoding flexibility by providing an additional decode window. This means that software now has the flexibility of assigning 2 different continuous SPI address spaces that use 2 different address decoding windows in the originating unit. Each of the 2 windows covers up to 4 CSs (of the total 8). The sub decoding, in this case, is accomplished via the <SPI Window0IndexLSB> field.

The SPI controller has a set of parameters to define guaranteed timing gaps, as listed below.

SPI CS Setup To ensure minimal gap between CS assertion and the first clock edge, set the <SPI_TCS_SETUP> field in the SPI Timing Parameters 1 Register (N=0–1) (Table 1332 p. 1077) to the required delay (in TCLK cycles).

SPI CS Hold To ensure a minimal gap between the last clock edge and the CS de-assertion, set the <SPI_TCS_HOLD> field in the SPI Timing Parameters 1 Register (N=0–1) (Table 1332 p. 1077) to the required delay (in TCLK cycles).

SPI CS De-select To ensure a minimal gap between the CS de-assertion and the next assertion, set the <SPI_TCSH_9_6> field and the <SP_TCSH> field in the SPI Timing Parameters 1 Register (N=0–1) (Table 1332 p. 1077) to the required delay (in TCLK cycles).

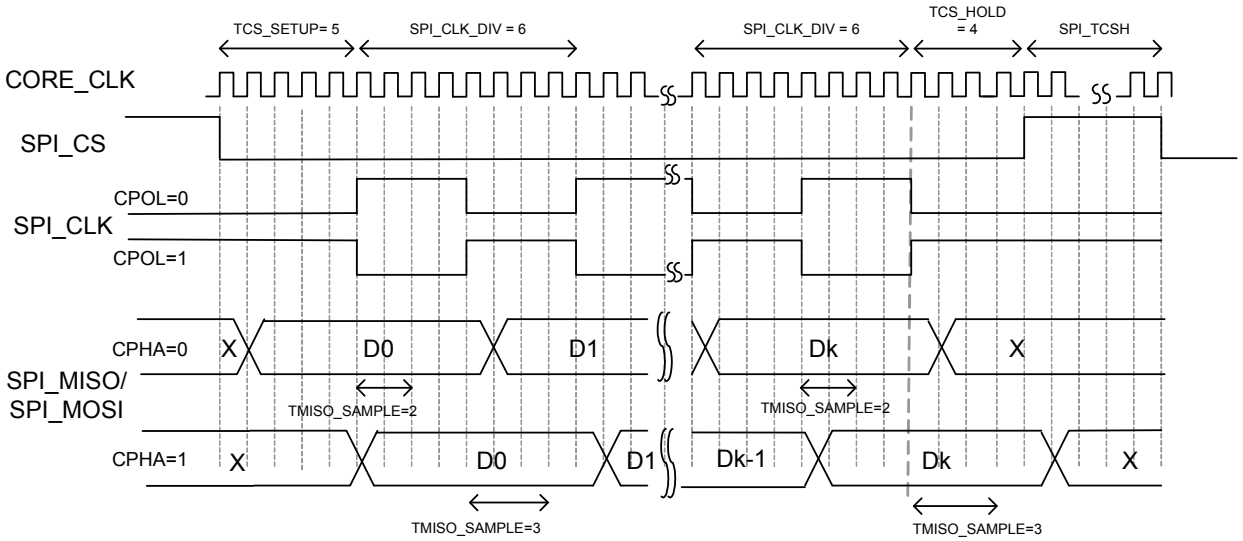
Typically, SPI<n>_MISO is sampled by the master in the sampling edge as defined by CPOL and CPHA modes. The internal sampling point of MISO can be delayed to relax the timing constraint for a high speed bus. To delay the sampling, configure the <SPI_TMISO_SAMPLE> field in the SPI Timing Parameters 1 Register (N=0–1) (Table 1332 p. 1077) to the required delay (in TCLK cycles), subject to the following constraints:

- The delay should be smaller then 1/2 SPI_CLK cycle.
- When CPHA=1, the delay should be smaller then SPI_TCS_HOLD.

Refer to Figure 72 for an example.

Document Classification: Proprietary Information

Figure 72: SPI Timing Example



22 Secure Digital Input/Output (SDIO) 3.0

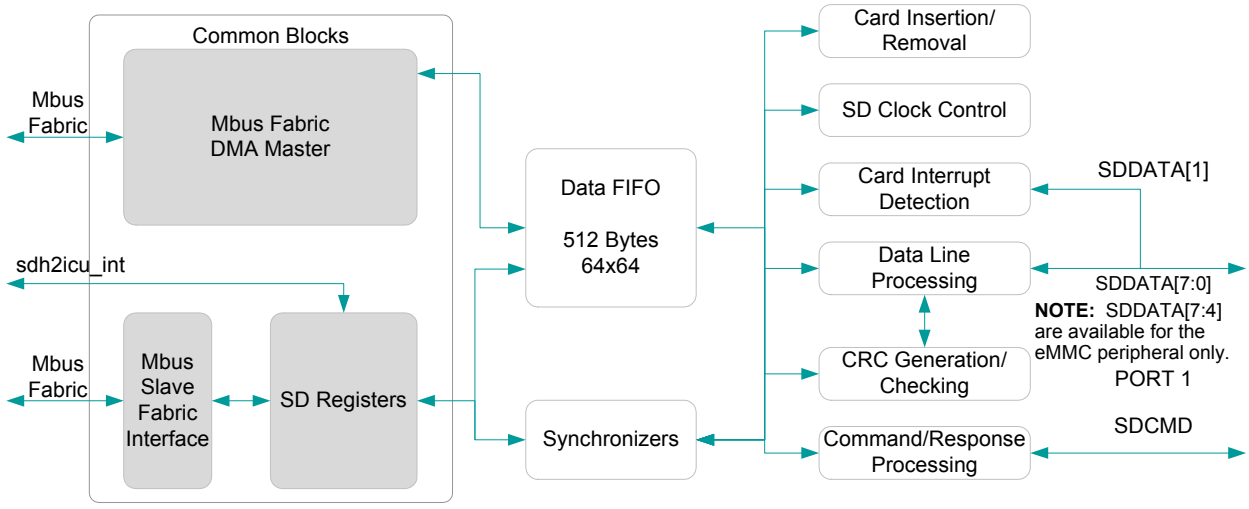
The SD host controller is a hardware block that acts as a host of the SD/MMC bus to transfer data between SD memory, MMC, SDIO cards, and the internal bus master. The SD host controller supports one socket.

One side of this block interfaces with an internal bus. The internal interface access can be set to the following modes:

- **Advanced DMA mode –** This block acts as an internal master that accesses the DMA descriptor list and performs the complete DMA according to the descriptor list (scatter gather list).
- **SDMA mode –** This block acts as an internal master which performs data transfer as programmed and generates an interrupt after completion of each data transfer.
- **PIO mode –** The CPU accesses the data through the internal bus. This interface is most likely used for debugging purposes.

The SDIO host controller block diagram is shown in Figure 73.

Figure 73: SDIO Host Block Diagram



The Secure Digital Input/Output (SDIO) 3.0 registers are located in Appendix A.17, Secure Digital Input/Output (SDIO) 3.0 Registers, on page 1082.

Copyright © 2015 Marvell

22.1 Features

- SD 3.0 and MMC 4.4.1 specification compliant
- Supports Advanced DMA 1 and Advanced DMA 2 (32-bit addressing)
- 1-bit/4-bit SDmem and SDIO
- 1-bit/4-bit/8-bit MMC
- Following modes are supported:
  - Default Speed (DS)–up to 25 MHz, 1.8V/3.3V signaling
  - High Speed (HS)–up to 50 MHz, 1.8V/3.3V signaling
  - SDR12–SDR up to 25 MHz, 1.8V/3.3V signaling
  - SDR25–SDR up to 50 MHz, 1.8V/3.3V signaling
  - SDR50–SDR up to 100 MHz, 1.8V signaling
  - DDR50–DDR up to 50 MHz, 1.8V/3.3V signaling
- Supports read wait control in SDIO cards
- Hardware generation/checking of CRC on all command and data transaction on the card bus
- True dual-port data buffer (64 x 64) used as internal buffer
- Suspend/resume in SDIO cards
- Card insertion/removal auto-detection using SDDATA[3].

Table 67: SD Feature Support

Feature	Supported
Supported SD Host Controller Specification	Version 3.0
Supported Physical Layer Specification	Version 3.01
Supported SDIO Specification	Version 3.0
VDDH	3.3 and 1.8V
Bus Interface Speed	UHS 100 MHz
SD bus width	1/4 bit
Wake-up Control	Yes
Suspend Resume	Yes
SDMA	Yes
ADMA	Yes
Data Transfer Block Sizes	Yes
IO_RW_DIRECT (CMD52)	Yes
IO_RW_EXTENDED (CMD53)	Yes
Suspend/Resume	Yes
Read Wait	Yes
Interrupt Period at the Data Block Gap in 4-bit SD Mode	Yes
Shared Bus	No
Programmable Clock Mode	No
UHS-I Host Type	SDR-VD
UHS-I Regulator Output	No
UHS-I Error Detection	No

Copyright © 2015 Marvell

Table 67: SD Feature Support (Continued)

Feature	Supported
CMD52 During Data Transfer	Yes
Power Selection/ Power State Control	Yes
Set Block Count (CMD23)	Yes
SDIO Multi Function	Yes
SDIO Combo	Yes

Note For more information on SD, refer to the SD specifications at http://www.sdcard.org.

22.2 MMC Mode

This mode is enabled by setting the <MMC_CARD> field in the CE-ATA Register (Table 1369 p. 1113) to 0x1. The rest of the data flow is similar to the regular SD mode.

22.3 Programming Guidelines

This section is intended to assist firmware development and is provided as a subset to the SD/MMC standard specifications.

22.3.1 Generation of 74 Clock Cycles After Reset

Some MMCs require that clock cycles be generated after reset. The host controller can be made to generate a programmable number of clock cycles. Use the following procedure for generating the cycle cycles:

1. Set the <MISC_INT_EN> field in the CE-ATA Register (Table 1369 p. 1114) to capture the completion event.
2. Set the <MISC_INT_INT_EN> field in the CE-ATA Register (Table 1369 p. 1114) to generate an interrupt for the completion event.
3. Configure the <GEN_PAD_CLK_CNT> field in the SD Extra Parameters Register (Table 1366 p. 1110) to the desired number of clock cycles. The default value is 74.
4. Set the <GEN_PAD_CLK_ON> field in the SD Extra Parameters Register (Table 1366 p. 1110).
5. Either wait for the interrupt (if enabled) or poll the <MISC_INT> field in the CE-ATA Register (Table 1369 p. 1114) to determine completion.

22.3.2 Enabling DDR50 Mode

For enabling DDR50 mode, set the <SDH_V18_EN> field and the <UHS_MODE_SEL> field in the Auto Host Control2 and Command12 Error Status Register (Table 1356 p. 1102).

Note For eMMC, the <1.8V Signaling Enable> field does not control the signaling in the I/O pads. At power up, the I/O pads auto-detect power at either 1.8 or 3.3 V.

Document Classification: Proprietary Information

22.3.3 Alternate Boot for MMCs

For alternate boot for MMCs, use the normal procedure as detailed in the SD specification. For more information on DDR50 mode, refer to Section 22.3.2, Enabling DDR50 Mode, on page 276. Program cmd0 as it would be with any data transfer command, that is, set the block count and the block size fields. In addition to those settings, there are a few additional settings that are required:

1. If the device has been enabled for DDR boot, set the <SDH_V18_EN> field and the <UHS_MODE_SEL> field in the Auto Host Control2 and Command12 Error Status Register (Table 1356 p. 1102).
2. If the device has been enabled to respond with boot ack, then set the <BOOT_ACK> field in the SD Extra Parameters Register (Table 1366 p. 1110) before issuing the cmd0.

For boot ack:

1. If the boot ack is not received within the programmed data timeout value, a data timeout interrupt will be generated (see the <DATA_TIMEOUT_ERR> field in the Normal and Error Interrupt Status Register (Table 1353 p. 1096)).
2. If the wrong boot ack code is received, then a CRC status error is generated (see the <ERR_INT> field in the Normal and Error Interrupt Status Register (Table 1353 p. 1096)).
3. If the correct boot ack is received and the read data start bit is not received in the specified time frame, then a data timeout interrupt will be generated.
4. If book ack is successful, then a transfer done interrupt will be generated.
   If a block by block status is required, then the <BLOCK_GAP_EVT> field in the Normal and Error Interrupt Status Register (Table 1353 p. 1097) should be set.

22.3.4 Dynamic Pad Clock Gating

By default, the host controller disables clock gating to the device. However, most newer versions of SD and MMC devices along with certain SDIO devices may support clock gating. In this case, set the <ASYNC_INT_EN> field in the Auto Host Control2 and Command12 Error Status Register (Table 1356 p. 1102).

22.3.5 ADMA2 Use with Large Block Count

This section describes use of ADMA2 when the block count is greater than 64K (65536 or 0xFFFF). For this scenario, software should perform the following:

1. Set the <MULTI_BLK_SEL> field in the Command and Transfer Mode Register (Table 1344 p. 1086) to 0x1.
2. Set the <BLK_CNT_DIS> field in the Command and Transfer Mode Register (Table 1344 p. 1086) field to 0x0.
3. Program the descriptor for the required data transfer length.
4. For this scenario, software must determine the end of a data transfer. To do so, enable DMA interrupts by setting bit [2] in the descriptor field to 0x1.

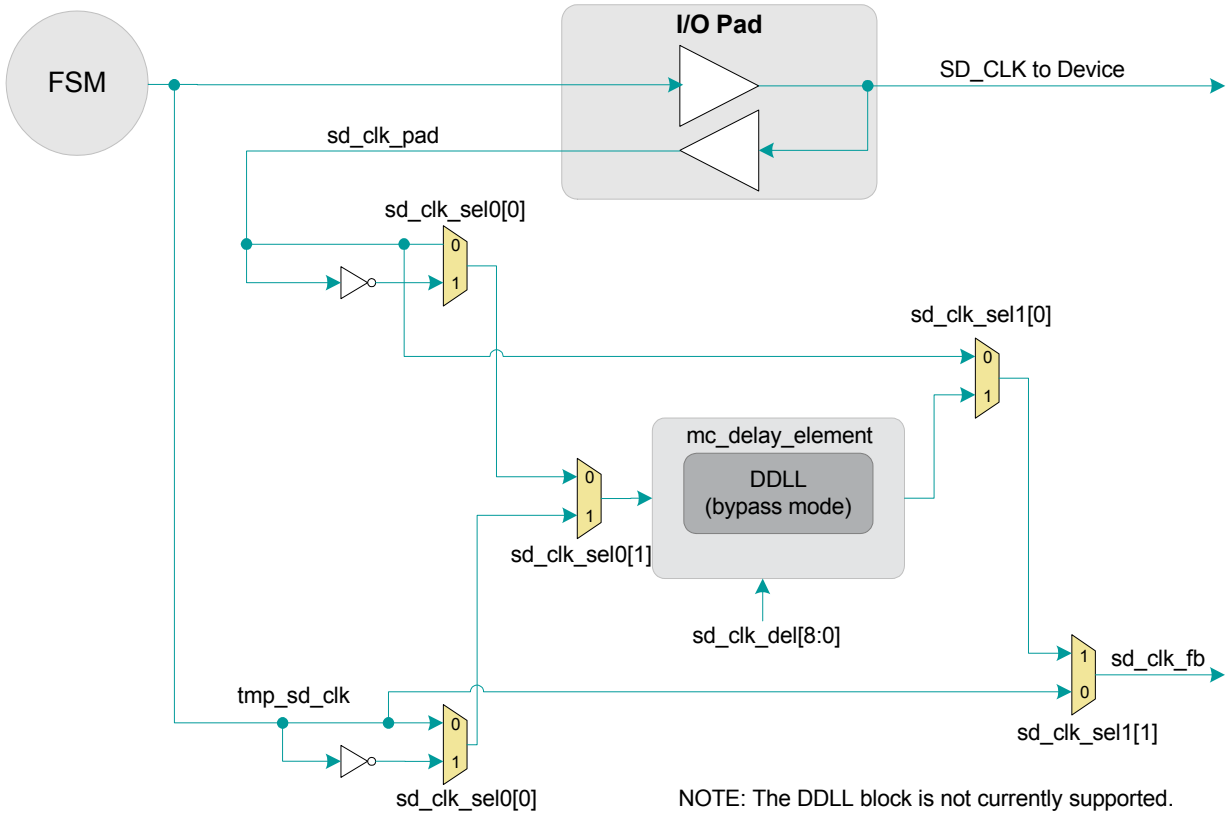
   NOTE: In this scenario, Auto CMD12 will not work since the host treats it as an infinite transfer.
5. Once software determines that the data transfer has completed (through receipt of the DMA_int and polling the data lines for idle state), one of the following will occur:
   - If CMD23 was not sent (either manually or automatically) prior to the data command, software must send CMD12 with the <CMD_TYPE> field in the Command and Transfer Mode Register (Table 1344 p. 1085) set to 0x3 (data abort) and then perform a software data reset by writing to the <SW_RST_DAT> field in the Timeout Control/Software Reset and Clock Control Register (Table 1352 p. 1093).
   - If CMD23 was sent (either manually or automatically) prior to the data command, software must perform a software data reset by writing 0x1 to perform a software data reset by writing to the <SW_RST_DAT>.

Copyright © 2015 Marvell

22.4 SD Clock Selection

Figure 74 shows the SD clock multiplexing structure for clock selection (sd_clk_fb) that is used to latch the input data and command. Refer to the <SDCLK_DELAY> field, the <SDCLK_SEL1>, and the <SDCLK_SEL0> field in the RX Configuration Register (Table 1371 p. 1115).

Figure 74: SD Clock Multiplexing Structure for Clock Selection (sd_clk_fb)



22.5 Power Up Sequence

Follow this procedure to power up the SDIO interface.

1. The <SDIO3 Clock Inv> field in the SDIO3 Configuration Register (Table 1908 p. 1409) should be set to 0x1 (Inverted) before starting power-up sequence.
2. Before switching speeds to SDR50 and while the SD_CLK is gated, it should be cleared to 0x0 (Not Inverted).
3. To switch to eMMC DDR, the following sequence should be applied before changing the <SD_FREQ_SEL_LO> field and the <SD_FREQ_SEL_HI> field in the Timeout Control/Software Reset and Clock Control Register (Table 1352 p. 1094) in the SDR-DDR switch, and while the SD_CLK is gated, clear the <SDIO3 Clock Inv> field to 0x0 (Not Inverted).
4. Select the new SD_CLK frequency by setting the <SD_FREQ_SEL_LO> and <SD_FREQ_SEL_HI> fields (the value must be greater than 0x4), and re-enable the SD_CLK.

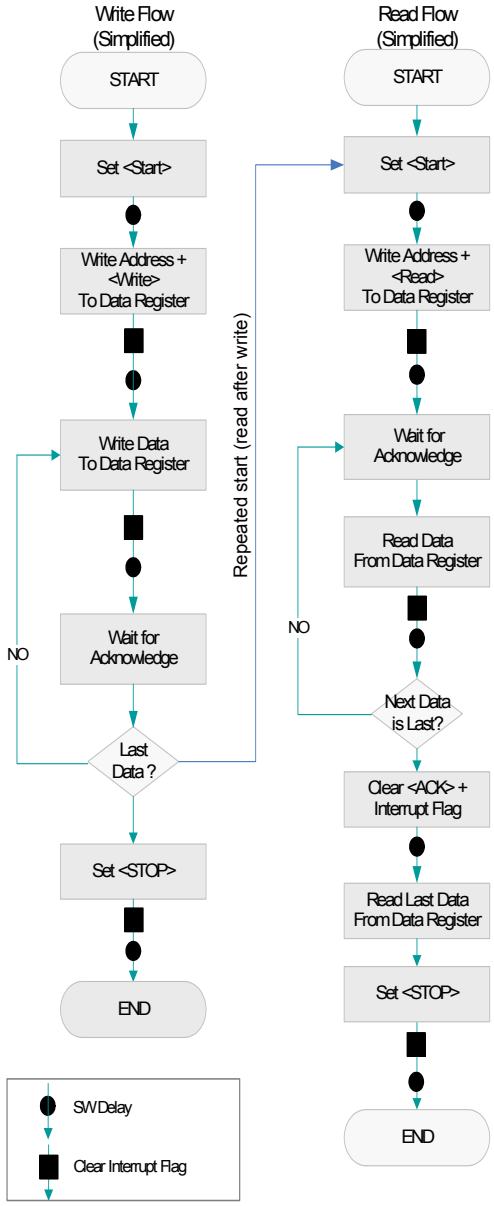
Copyright © 2015 Marvell

23 Inter-Integrated Circuit Interface (I2C)

This section describes the device's two general-purpose Inter-Integrated Circuit Interface (I2C) ports. Each port can act as an I2C master or as a slave. Figure 75 provides a high-level (simplified) I2C transaction flow diagram. For full details, refer to Section 23.3, I2C Functional Description.

If enabled via reset strap, port0 also supports initialization from an external I2C serial ROM.

Figure 75: I2C Transaction Flow Diagram



The I^2C registers are located in Appendix A.18, Inter-Integrated Circuit (I2C) Registers, on page 1120.

23.1 Features

The I^2C interface supports the following features:

- Two I^2C ports
- S-Mode (up to 100 kHz SCL clock rate)
- General purpose I^2C master/slave
- Transaction level I^2C master activation
- EEPROM Serial initialization support

23.2 I^2C Bus Operation

The I^2C port consists of 2 open drain signals:

- SCL (Serial Clock)
- SDA (Serial address/data)

The I^2C master starts a transaction by driving a start condition followed by a 7- or 10-bit slave address and a read/write bit indication. The target I^2C slave responds with acknowledge.

In case of a write access (R/W bit is 0 following the I^2C slave acknowledge), the master drives eight bits of data, and the slave responds with acknowledge. This write access (8-bit data followed by acknowledge) continues until the I^2C master ends the transaction with a stop condition.

In case of a read access following the I^2C slave address acknowledge, the I^2C slave drives eight bits of data and the master responds with acknowledge. This read access (8-bit data followed by acknowledge) continues until the I^2C master ends the transaction by responding with no acknowledge to the last 8-bit data, followed by a stop condition.

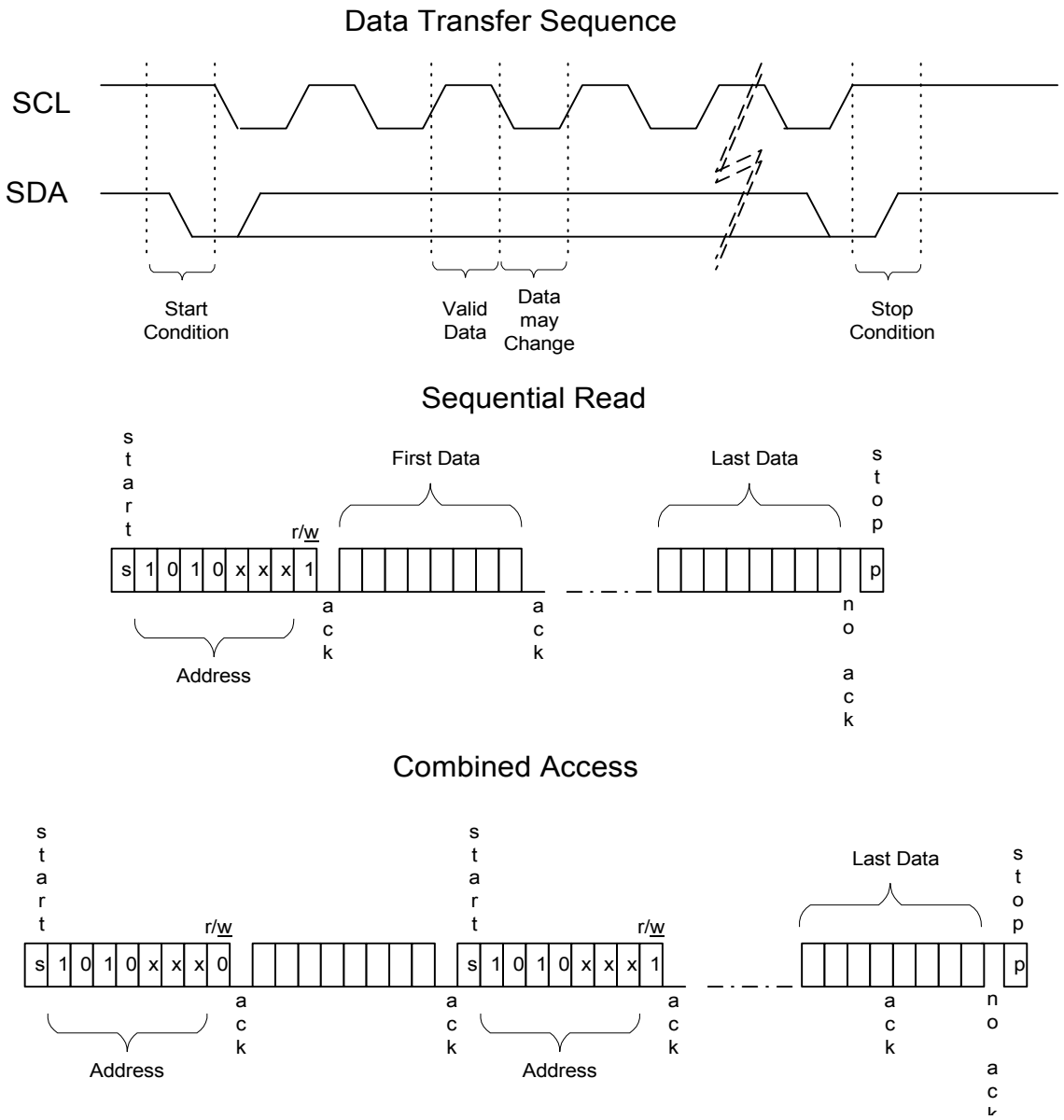
A target slave that cannot drive valid read data right after it received the address, can insert “wait states” by forcing SCL low until it has valid data to drive on the SDA line.

A master is allowed to combine 2 transactions. After the last data transfer, it can drive a new start condition followed by new slave address, rather than drive stop condition. Combining transactions guarantees that the master does not lose arbitration to some other I^2C master.

I^2C examples are shown in Figure 76.

Document Classification: Proprietary Information

Figure 76: I^2C Examples



23.3 I^2C Functional Description

The port can act as master, generating read/write requests, and as a slave, responding to read/write requests from an external master. It can be used for various applications, and can control other I^2C on-board devices such as temp sensors, to read the SPD ROM. It is also used for serial ROM initialization (see the Serial ROM Initialization section in the device *Hardware Specifications*).

The I^2C interface master and slave activities are handled by a simple CPU access to internal registers, plus an interrupt interface. The protocol is byte oriented— the core processor is required to handle each transmitted/received byte. The following sections describe I^2C registers and receive/transmit operation.

Note

The I²C port is also used for Serial ROM initialization (if enabled). The CPU must not access I²C port registers during the initialization period.

23.3.1 I²C Slave Address Registers

The I²C slave interface supports both 7-bit and 10-bit addressing. The slave address is programmed by the I2C Slave Address Register (t=0–3) (Table 1383 p. 1121).

When the I²C receives a 7-bit address after a start condition, it compares that address against the value programmed in the Slave Address register. If the address matches, it responds with acknowledge.

If the received 7 address bits are '11110xx', meaning that it is an 10-bit slave address, the I²C compares the received 10-bit address with the 10-bit value programmed in the Slave Address and Extended Slave Address registers. If the address matches, it responds with acknowledge.

The I²C interface also supports slave response to general call transactions.

23.3.2 I²C Data Register

The 8-bit data register is used both in master and slave modes.

In master mode, the core processor must place the slave address or write data to be transmitted. In case of read access, it contains received data (need to be read by the core processor).

In slave mode, the Data register contains data received from master on write access, or data to be transmitted (written by the core processor) on read access.

Note

The data register Most Significant bit (MSb) contains the first bit to be transmitted or being received.

23.3.3 I²C Control Register

This 8-bit register contains the following bits:

Table 68: I²C Control Register Bits

Bit	Function	Description
1:0	Reserved	Read only 0.
2	Acknowledge Bit	When set to 1, the I²C drives an acknowledge bit on the bus in response to a received address (slave mode), or in response to a data received (read data in master mod, write data in slave mode). For a master to signal a I²C target a read of last data, the core processor must clear this bit (generating no acknowledge bit on the bus). For the slave to respond, this bit must always be set back to 1.
3	Interrupt Flag	If any of the interrupt events occur, set to 1 by I²C hardware If set to 1 and I²C interrupts are enabled through bit[7], an interrupt is asserted.

Copyright © 2015 Marvell

Table 68: I2C Control Register Bits (Continued)

Bit	Function	Description
4	Stop Bit	When set to 1, the I2C master initiates a stop condition on the bus. The bit is set only. It is cleared by I2C hardware after a stop condition is driven on the bus.
5	Start Bit	When set to 1, the I2C master initiates a start condition on the bus, when the bus is free, or a repeated start condition, if the master already drives the bus. The bit is set only. It is cleared by I2C hardware after a start condition is driven on the bus.
6	I2C Enable	If set to 1, the I2C slave responds to calls to its slave address, and to general calls if enabled. If cleared to 0, SDA and SCL inputs are ignored. The I2C slave does not respond to any address on the bus.
7	Interrupt Enable	If set to 1, I2C interrupts are enabled. Marvell recommends to use the I2C interrupt to interface the I2C module, rather than using the register polling method.
31:8	Reserved	Reserved

23.3.4 I2C Status Register

This 8-bit register contains the current status of the I2C interface. Bits[7:3] are the status code, bits[2:0] are Reserved (read only 0). Table 69 lists all possible status codes.

Table 69: I2C Status Codes

Code	Status
0x00	Bus error. **NOTE:** A bus error occurs if the I2C slave interface drives the bus (Data or Clock) when it should not. To recover from this error, set the <Stop> field in the I2C Control Register (t=0–3) (Table 1385 p. 1122) and clear the interrupt.
0x08	Start condition transmitted.
0x10	Repeated start condition transmitted.
0x18	Address + write bit transmitted, acknowledge received.
0x20	Address + write bit transmitted, acknowledge not received.
0x28	Master transmitted data byte, acknowledge received.
0x30	Master transmitted data byte, acknowledge not received.
0x38	Master lost arbitration during address or data transfer.
0x40	Address + read bit transmitted, acknowledge received.
0x48	Address + read bit transmitted, acknowledge not received.
0x50	Master received read data, acknowledge transmitted.
0x58	Master received read data, acknowledge not transmitted.

Table 69: I²C Status Codes (Continued)

Code	Status
0x60	Slave received slave address, acknowledge transmitted.
0x68	Master lost arbitration during address transmit, address is targeted to the slave (write access), acknowledge transmitted.
0x70	General call received, acknowledge transmitted.
0x78	Master lost arbitration during address transmit, general call address received, acknowledge transmitted.
0x80	Slave received write data after receiving slave address, acknowledge transmitted.
0x88	Slave received write data after receiving slave address, acknowledge not transmitted.
0x90	Slave received write data after receiving general call, acknowledge transmitted.
0x98	Slave received write data after receiving general call, acknowledge not transmitted.
0xA0	Slave received stop or repeated start condition.
0xA8	Slave received address + read bit, acknowledge transmitted.
0xB0	Master lost arbitration during address transmit, address is targeted to the slave (read access), acknowledge transmitted.
0xB8	Slave transmitted read data, acknowledge received.
0xC0	Slave transmitted read data, acknowledge not received.
0xC8	Slave transmitted last read byte, acknowledge received.
0xD0	Second address + write bit transmitted, acknowledge received.
0xD8	Second address + write bit transmitted, acknowledge not received.
0xE0	Second address + read bit transmitted, acknowledge received.
0xE8	Second address + read bit transmitted, acknowledge not received.
0xF8	No relevant status. Interrupt flag is kept 0.

23.3.5 Baud Rate Register

The I²C module contains a clock divider to generate the SCL clock. Setting the <N> field and the <M> field in the I2C Baud Rate Register (t=0–3) (Table 1386 p. 1123) defines SCL frequency (F_{SCL}) as follows:

$$F_{SCL} = \frac{F_{TClk}}{10 \cdot (M+1) \cdot 2^{(N+1)}}$$

Copyright © 2015 Marvell

Note Where M is the value represented by bits[6:3] and N is the value represented by bits[2:0]: If for example, M=4 and N=5 (which are the default values), running TCLK at 200 MHz (F_{TClk}) results in a SCL frequency of 62.5 kHz.

As defined in the I^2C specifications for S-Mode, the maximum supported SCL frequency is 100 kHz (Standard mode), this speed is supported on all ports. In addition, for I^2C Port0, this device also supports Fast mode with an SCL frequency of 400 kHz. Set the SCL frequency using the <I2C Select> field in the IO Configuration 4 Register (Table 1918 p. 1414).

The Baud Rate register must be set properly, even when using the I^2C port as a slave only. It should be set such that F_{SCL} will be in the range of x1 to x2 of the I^2C bus frequency.

23.3.6 I^2C Port Master Operation

Refer also for Figure 75 to illustrate the transaction flow. In that figure, SW delay represent a delay of at least 2 internal clock cycles (x10 SCL clock rate, at 100 kHz SCL, for example, the delay is 2 µs).

A master write access consists of the following steps:

1. The core processor sets the <Start> field in the I2C Control Register (t=0–3) (Table 1385 p. 1122) to 1. The I^2C master then generates a start condition as soon as the bus is free, sets an Interrupt flag, and sets the Status register to 0x8.
2. The core processor writes a 7-bit address plus a write bit to the Data register and clears the Interrupt flag for the I^2C master interface to drive the slave address on the bus. The target slave responds with acknowledge. This causes an Interrupt flag to be set and a status code of 0x18 is registered in the Status register.

   If the target I^2C device has an 10-bit address, the core processor needs to write the remainder 8-bit address bits to the Data register. The core processor then clears the Interrupt flag for the master to drive this address on the bus. The target device responds with acknowledge, causing an Interrupt flag to be set and status code of 0xD0 be registered in the Status register.
3. The core processor writes a data byte to the Data register, and then clears the Interrupt flag for the I^2C master interface to drive the data on the bus. The target slave responds with acknowledge, causing the Interrupt flag to be set, and status code of 0x28 be registered in the Status register. The core processor continues this loop of writing new data to the Data register and clearing the Interrupt flag as long as it needs to transmit write data to the target.
4. After the last data transmit, the core processor may terminate the transaction or restart a new transaction. To terminate the transaction, the core processor sets the Control Register <Stop> bit and then clears the Interrupt flag. This causes the I^2C master to generate a stop condition on the bus and to return to idle state. To restart a new transaction, the core processor sets the <Start> field in the I2C Control Register (t=0–3) (Table 1385 p. 1122) and clears the Interrupt flag, thus causing the I^2C master to generate a new start condition.

Note The above sequence describes a normal operation. There are also abnormal cases, such as a slave not responding with acknowledge or arbitration loss. Each of these cases is reported in the Status register and needs to be handled by the core processor.

A master read access consists of the following steps:

1. Generating start condition, exactly the same as in the case of write access (see Section 23.3.1, I^2C Slave Address Registers).

2. Drive 7- or 10-bit slave address, exactly the same as in the case of write access, with the exception that the status code after the first address byte transmit is 0x40, and after 2nd address byte transmit (in case of 10-bit address) is 0xE0.
3. Read data being received from the target device is placed in the data register and acknowledge is driven on the bus. Also interrupt flag is set, and status code of 0x50 is registered in the Status register. The core processor reads data from Data register and clears the Interrupt flag to continue receiving next read data byte. This loop is continued as long as the core processor wishes to read data from the target device.
4. To terminate, the read access needs to respond with no acknowledge to the last data. It then generates a stop condition or generates a new start condition to restart a new transaction. With last data, the core processor clears the <ACK> field in the I2C Control Register (t=0–3) (Table 1385 p. 1122) (when clearing the Interrupt bit), causing the I²C master interface to respond with no acknowledge to last received read data. In this case, the Interrupt flag is set with status code of 0x58. Now, the core processor can issue a stop condition or a new start condition.



Note

This sequence describes a normal operation. There are also abnormal cases, such as the slave not responding with acknowledge, or arbitration loss. Each of these cases is reported in the Status register and needs to be handled by core processor.

23.3.7 I²C Port Slave Operation

The I²C slave interface can respond to a read access, driving read data back to the master that initiated the transaction, or respond to write access, receiving write data from the master.

Upon detecting a new address driven on the bus with a read bit indication, the I²C slave interface compares the address against the address programmed in the Slave Address register. If it matches, the slave responds with acknowledge. It also sets the Interrupt flag, and sets the status code to 0xA8.



Note

If the I²C slave address is 10-bit, the interrupt flag is set, and the status code changes only after receiving and identifying an address match on the second address byte.

The core processor now must write new read data to the Data register and clear the Interrupt flag, causing I²C slave interface to drive the data on the bus. The master responds with acknowledge causing an Interrupt flag to be set, and status code of 0xB8 to be registered in the Status register.

If the master does not respond with acknowledge, the Interrupt flag is set, a status code 0f 0xC0 is registered, and the I²C slave interface returns back to idle state.

If the master generates a stop condition after driving an acknowledge bit, the I²C slave interface returns back to idle state.

Upon detecting a new address driven on the bus with write bit indication, the I²C slave interface compares the address against the address programed in the Slave Address register. If the address matches, it responds with acknowledge.matches, responds with acknowledge. It also sets an Interrupt flag, and sets the status code to 0x60 (0x70 in case of general call address, if general call is enabled).

Following each write byte received, the I²C slave interface responds with acknowledge, sets an Interrupt flag, and sets status code to 0x80 (0x90 in case of general call access). The core processor then reads the received data from Data register and clears Interrupt flag to allow transfer to continue.

Copyright © 2015 Marvell

If a stop condition or a start condition of a new access is detected after driving the acknowledge bit, an Interrupt flag is set and a status code of 0xA0 is registered.

23.3.8 I2C Port Transaction Generator

To reduce software overhead each I2C port can be activated in a transaction level mode. In this mode, all properties of an I2C Master-Receive or an I2C Master-Transmit are pre-configured. Following the transaction's properties configuration, the transaction generator is signaled to start. When the transaction is completed, an interrupt is asserted.

23.3.8.1 I2C Transaction Generation – Master-Transmitter

The following sequence performs an I2C Master-Transmitter transaction:

1. Set the <Wr> and clear the <Rd> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127) in the same register.
2. Define the slave address length by writing (0x0 = 7-bit address or 0x1 = 10-bit address) to the <ExtendAddr> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127).
3. Write the slave address of the peer I2C to the <Addr> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127).
4. Write the number of data bytes to be transmitted to the <TxSize> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127).
5. Write the data to be transmitted to the <TxDataLow> field in the I2C Bridge Transmit Data Low Register (t=0–3) (Table 1392 p. 1126) and the <TxDataHigh> field in the I2C Bridge Transmit Data High Register (t=0–3) (Table 1393 p. 1126).
6. Set the <BridgeEn> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1126) to start the transaction.
7. When the transaction is completed an I2C interrupt is set.
8. Read the <Error> field in the I2C Bridge Status Register (t=0–3) (Table 1397 p. 1128) to validate that transmission has been completed successfully.

23.3.8.2 I2C Transaction Generation – Master-Receiver

The following sequence performs an I2C Master-Receiver transaction:

1. Set the <Rd> field and clear the <Wr> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127).
2. Define the slave address length by writing to the <ExtendAddr> (0x0 = 7-bit address or 0x1 = 10-bit address).
3. Write to the <Addr> field in the slave address of the peer I2C.
4. Write the number of data bytes to be received in the <RxSize> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127).
5. Set the <BridgeEn> field to start the transaction.
6. When the transaction is completed an I2C interrupt is set.
7. Read the <Error> to validate that data has been received successfully.
8. Read the requested data from the <RxDataLow> field in the I2C Bridge Receive Data Low Register (t=0–3) (Table 1394 p. 1126) and the <RxDataHigh> field in the I2C Bridge Receive Data High Register (t=0–3) (Table 1395 p. 1126).

Copyright © 2015 Marvell

23.3.8.3 I²C Transaction Generation – Master-Transmitter Followed by Master-Receiver

The following sequence performs an I²C Master-Transmitter transaction that is followed by a Master-Receiver transaction:

1. Set both the <Rd> field and the <Wr> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1127).
2. Define the delimiter between the Master-Transmitter phase and the Master-Receiver phase by writing the required type of Start transmission (0 = Stop followed by Start or 1 = Repeated Start) to the <RepeatedStart> field in the I2C Bridge Control Register (t=0–3) (Table 1396 p. 1126).
3. Define the slave address length by writing to the <ExtendAddr> field (0x0 = 7-bit address or 0x1 = 10-bit address).
4. Write the slave address of the peer I²C to the <Addr> field.
5. Write the number of data bytes to be transmitted during the Master-Transmitter phase to the <TxSize>.
6. Write the data to be transmitted at the Master-Transmitter phase to the <TxDataLow> field in the I2C Bridge Transmit Data Low Register (t=0–3) (Table 1392 p. 1126) and the <TxDataHigh> field in the I2C Bridge Transmit Data High Register (t=0–3) (Table 1393 p. 1126).
7. Write the number of data bytes to be received during the Master-Receiver phase to the <RxSize>.
8. Set the <BridgeEn> field to start the transaction.
9. When the transaction is completed an I²C interrupt is set.
10. Read the <Error> field in the I2C Bridge Status Register (t=0–3) (Table 1397 p. 1128) to validate that both phases have completed successfully.
11. Read the requested data from the <RxDataLow> field in the I2C Bridge Receive Data Low Register (t=0–3) (Table 1394 p. 1126) and the **<RxDataHigh>** field in the I2C Bridge Receive Data High Register (t=0–3) (Table 1395 p. 1126).

23.3.8.4 I²C Transaction Generation Example – Reading from EPROM

To demonstrate usage of a master-transmitter followed by a master-receiver, the following sequence is an example that performs a read from a standard I²C EPROM:

1. <Rd>=1, <Wr>=1.
2. <RepeatedStart>=1.
3. <ExtendAddr>=0.
4. <TxSize>=2.
5. <TxDataLow> and <TxDataHigh>=The byte address in the EPROM.
6. <RxSize>=8.
7. <Addr>=0x50 // 7 bit slave address of the EPROM.
8. Set the <BridgeEn> field to start the transaction.
9. When the transaction is completed an I²C interrupt is set.
10. Read the <Error> field in the I2C Bridge Status Register (t=0–3) (Table 1397 p. 1128) to validate that both phases have completed successfully.
11. Read the 8 bytes from the <RxDataLow> and <RxDataHigh> fields.

23.3.8.5 I²C Transaction Generation Timing Gaps

Timing gaps with in an auto generated transaction can be configured using the I2C Bridge Timing Gaps Register (t=0–3) (Table 1400 p. 1129).

Copyright © 2015 Marvell
Document Classification: Proprietary Information

23.4 I2C Serial ROM Initialization

I2C port0 can be set at reset to perform serial ROM initialization (act as a I2C master, reading data from external I2C ROM, and writing this data to the chip registers). For further details about serial ROM initialization, refer to the Reset section in the device Hardware Specifications.

23.5 I2C Hardware Slave

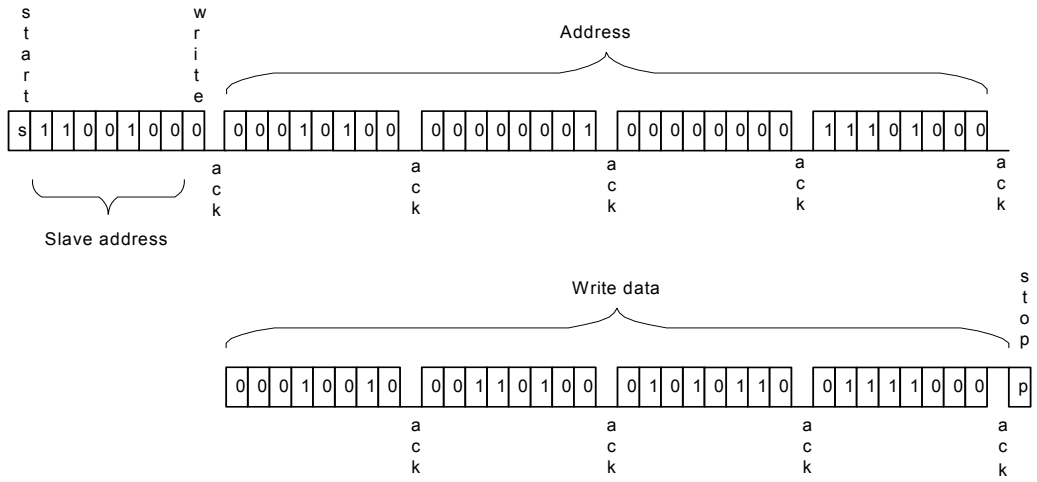
I2C Port1 can be set at reset to act as a slave hardware port.

The hardware slave is a an internal DMA that can receive read and write transactions from the I2C bus, and issues them as internal requests within the device. It then completes the transaction over the I2C port. These operations do not require software intervention.

As a hardware slave port, I2C Port1 corresponds to a fixed 7-bit address of 'b1100100.

A word write example is shown in Figure 77.

Figure 77: I2C Debug Port Write Example

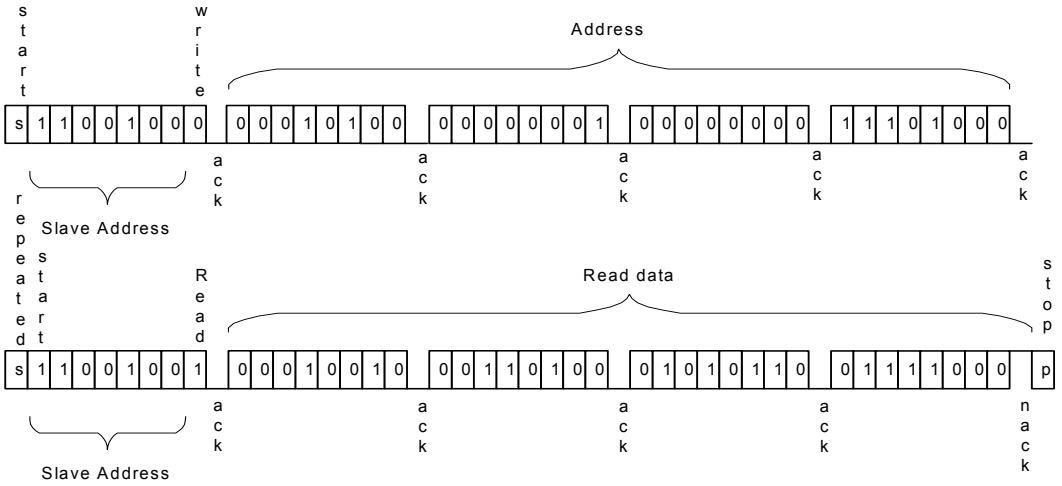


The write sequence is as follows:

1. The external master drives the start bit followed by the slave address (7'b1100100) with the read/write bit cleared to 0 (write). The device responds with acknowledge (ACK).
2. The external master drives 4 bytes that represent the 32-bit address within the device address space (0x1401.00E8 in the above example). The device responds with an acknowledgement to each byte.
3. The external master drives 4 bytes of data (0x1234.5678 in the above example). The device responds with ACK to all 4 bytes.
4. The external master drives a stop bit.
5. The device I2C hardware slave writes the 32-bit data to the location pointed to by the 32-bit address.

A word read example is shown in Figure 78

Figure 78: I²C Debug Port Read Example



The read sequence is as follows:

1. The external master drives the start bit followed by the slave address (7'b1100100) with read/write bit cleared to 0 (write). The device responds with acknowledge (ack).
2. The external master drives 4 bytes that represent a 32-bit address within the device address space (0x1401.00E8 in the above example). The device responds with ACK to each byte.
3. The external master drives a new start bit followed by slave address (7'b1100100) with read/write bit set to 1 (read). The device responds with acknowledge.
4. The device I²C hardware slave controller reads 32-bit of data from the location pointed by the 32-bit address.
5. The device I²C hardware slave controller drives the 4 bytes received from the internal register over the data of the master-receive transaction (0x1234.5678 in the above example). The external master responds with ACK to the first 3 bytes and NACK to the last byte.
6. The external master drives a stop bit.

This sequence is a combined sequence. The external master drives the start bit of the read access right after the acknowledge bit of last write access without a STOP in between. I²C hardware slave also tolerates a non-combined access, in which the external master generates a stop bit at the end of the write access, and follows it by issuing the read access.

Copyright © 2015 Marvell

24 Universal Asynchronous Receiver/Transmitter (UART) Interface

The device supports the UART interface through the UA0/1_TXD and UA0/1_RXD pins and provides modem control functions through the UA0/1_CTS and UA0/1_RTS pins.

The UART port performs the following data conversions:

- Serial-to-parallel conversion on data characters received from a peripheral device or a modem
- Parallel-to-serial conversion on data characters received from the device

The device can read the complete UART status from the Line Status (LSR) Register (n=0–3) (Table 1411 p. 1136). Status information includes the type and condition of transfer operations and error conditions (parity, overrun, framing, or break interrupt) associated with the UART.

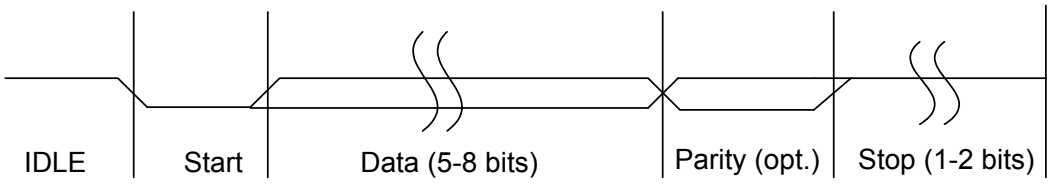
The serial port operates in either FIFO or Standard mode. In addition, for transmit operations, the UART can operate in DMA-Based mode.

- In FIFO mode, a 16-byte transmit FIFO buffer holds data from the device until it is transmitted on the serial link; a 16-byte FIFO buffer receives data from the serial link until it is read by the device.
- In Standard mode, an 8-bit transmit buffer holds data from the device until it is transmitted on the serial link; an 8-bit buffer receives data from the serial link until it is read by the device.

The device supports the UART interface through the UA0/1_TXD and UA0/1_RXD pins and provides modem control functions through the UA0/1_CTS and UA0/1_RTS pins.

Figure 79 shows a UART character format on the serial line.

Figure 79: UART Character Format

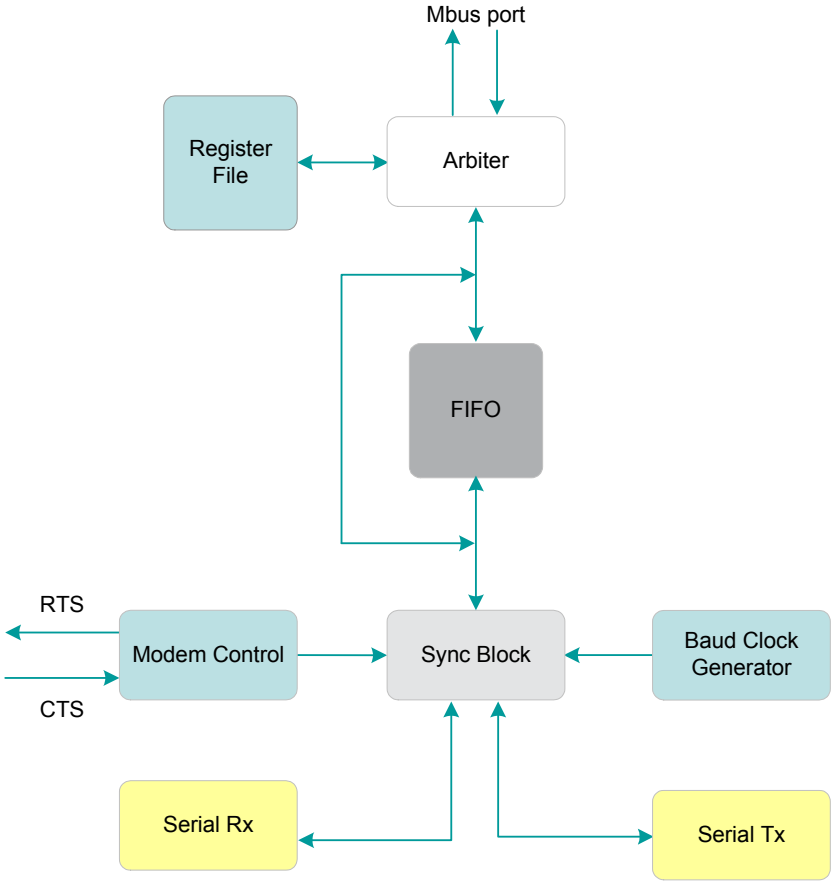


The UART is 16750 compatible with busy functionality. For details about this functionality, see the UART registers.

The UART registers are located in Appendix A.19, Universal Asynchronous Receiver/Transmitter (UART) Registers, on page 1130.

Figure 80 presents the main blocks of the UART controller.

Figure 80: UART Controller Block Diagram



24.1 Features

The UART interface incorporates the following features:

- Ability to add or delete standard asynchronous communications bits (start, stop, and parity) in the serial data
- Independently controlled transmit, receive, line-status, and data-set interrupts
- Programmable baud-rate generator (see Section 24.3, Programmable Baud-Rate Generator)
- Modem flow control functions (CTS and RTS).
- Programmable serial interface:
  - 5-, 6-, 7- or 8-bit characters
  - Even, odd, or no parity detection
  - 1 or 2 stop-bit generation
- 16-byte transmit FIFO
- 16-byte receive FIFO
- Complete status-reporting capability
- Ability to generate and detect line breaks

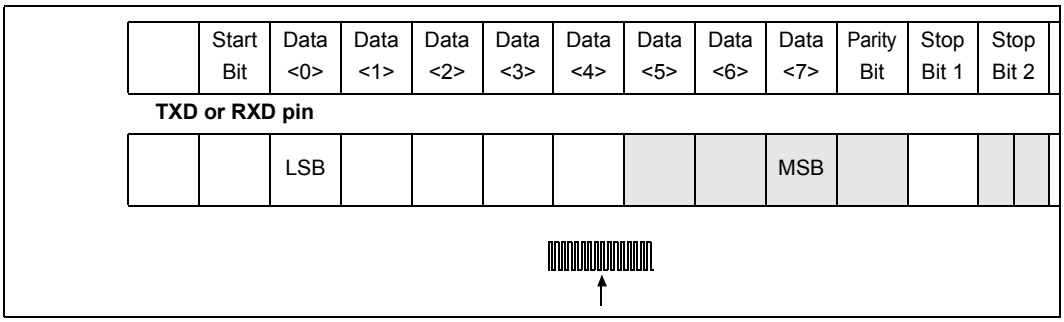
Copyright © 2015 Marvell

- Internal diagnostic capabilities that include:
  - Loopback controls for communication link fault isolation
  - DMA-based UART
  - Break, parity, and framing-error simulation
- Fully prioritized interrupt system controls

24.2 Functional Description

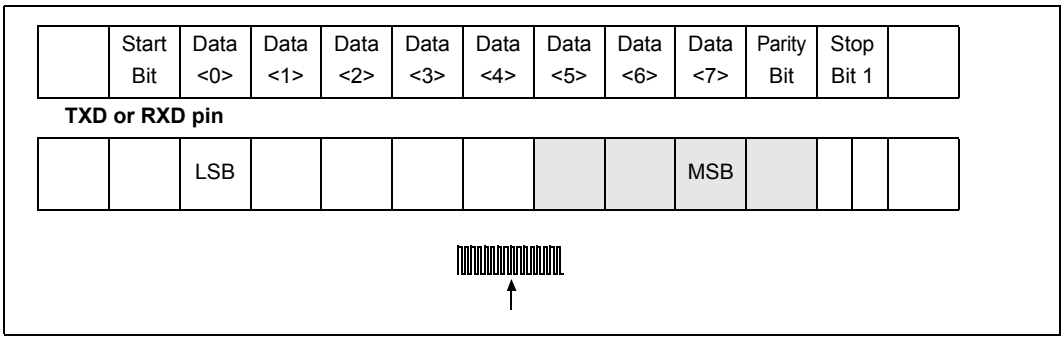
Figure 81 and Figure 82 shows the format of a UART data frame, respectively, with two stop bits and with one stop bit.

Figure 81: UART Character (Two Stop Bits Example)



To ensure bus stability, the receiver samples the serial input data at approximately the mid-point of the bit sequence, once the start bit has been detected.

Figure 82: UART Character (One Stop Bit Example)



The receive-data sample-counter frequency is 16 times the value of the bit frequency. The 16x clock is created by the baud-rate generator. Each bit is sampled three times in the middle of the bit sequence. Shaded bits are optional and can be programmed by software.

The data frame is between 7 and 12 bits long, depending on the size of the data programmed, parity status (enabled/disabled), and the number of stop bits. A data frame begins by transmitting a start bit that is represented by a high to low transition. The start bit is followed by five to eight bits of data that begin with the least significant bit (LSB).

The data bits are followed by an optional parity bit. The parity bit is set if even parity is enabled and the data byte has an odd number of ones, or if odd parity is enabled and the data byte has an even number of ones. The data frame ends with one or two stop bits, as programmed by software. The stop bits are represented by one or two successive bit periods of logic one.

Copyright © 2015 Marvell

Each UART port has a transmit and a receive FIFO.

- The transmit FIFO is 16 bytes deep.
- The receive FIFO is 16 bytes deep.

24.3 Programmable Baud-Rate Generator

Each UART port includes a programmable baud-rate generator that can take a fixed-input clock and divide it to generate the preferred baud rate. The baud rate is calculated by taking the TCLK frequency and dividing it by a value between 1 and (2^{16} - 1), to produce a 16x clock that is used to drive the internal transmit and receive logic. Each UART operates in an environment that is either controlled by the software and can be polled, or is interrupt driven.

The baud-rate generator output frequency is 16 times the baud rate. The <DivLatchLow> field in the Divisor Latch Low (DLL) Register (n=0–3) (Table 1402 p. 1131) and the <DivLatchHigh> field in the Divisor Latch High (DLH) Register (n=0–3) (Table 1406 p. 1133) make up the two 8-bit divisor latch fields that store the divisor in a 16-bit binary format. Load these divisor latches during initialization to ensure that the baud-rate generator operates properly. The 16x clock stops if both fields are loaded with 0x0.

The baud rate of the data shifted into or out of a UART is given by the formula:

$$\mathbf{BaudRate} = \frac{\text{TCLK freq}}{(16 \times \text{Divisor})}$$

Table 70: Typical Baud Rates, TCLK = 200 MHz

Required Baud	Divisor (decimal)	Actual Baud	Error (%)
1200	10,417	1200	0
2400	5208	2400	0
4800	2604	4800	0
9600	1302	9600	0
19,200	651	19,200	0
38,400	326	38,344	0.1
57,600	217	57,604	0.1
76,800	163	76,687	0.1
115,200	109	114,679	0.5
230,400	54	231,481	0.5
460,800	27	462,963	0.5
500,000	25	500,000	0
921,600	14	892,857	3.1
1,000,000	13	961,538	3.8
1,382,400	9	1,388,889	0.5

Copyright © 2015 Marvell

Table 70: Typical Baud Rates, TCLK = 200 MHz (Continued)

Required Baud	Divisor (decimal)	Actual Baud	Error (%)
1,500,000	8	1,562,500	4.2

Table 71: Typical Baud Rates, TCLK = 250 MHz

Required Baud	Divisor (decimal)	Actual Baud	Error (%)
1200	13,021	1200	0
2400	6510	2400	0
4800	3255	4800	0
9600	1628	9600	0
19,200	814	19,200	0
38,400	407	38,400	0
57,600	271	57,657	0.1
76,800	203	76,970	0.2
115,200	136	114,890	0.27
230,400	68	229,779	0.27
460,800	34	459,559	0.27
500,000	31	504,032	0.81
921,600	17	919,118	0.27

24.4 UART Debug Mode

When the BootROM device is the UART, the applicable UART Debug mode commands are:

- Read: r address (example r 0xF1040000)
- Write: w address value
- Execute: x bootmode (0x1..0x1F)
- Jump: j address

25 General Purpose Input/Output Ports

This section describes the device GPIO interface.

The GPIO registers are described in Appendix A.20, General Purpose Input/Output (GPIO) Registers, on page 1140.

25.1 Features

The device GPIO interface includes the following features:

- GPIO pin can be set as input or output.
- GPIO inputs are latched and can also be used to register and generate maskable interrupts from external devices.
- GPIO outputs can be configured to the desired driving value (High or Low) and can follow an open drain functionality.
- As GPIO outputs, these pins can be used for driving LEDs or Pulse Width Modulation (PWM) controlled devices, such as fans, with programmable ON/blinking rates and durations.

25.2 Functional Description

The device supports a General Purpose IO (GPIO) interface. Each of the GPIO pins can be assigned to act as a general purpose input or output pin and can be used to register external interrupts (when assigned as an input pin). The usage model of input, output and the value can be configured via configuration registers mentioned in the following section. The GPIO interface is multiplexed over the MPP pins. Refer to the device hardware specifications for more details on the multiplexing pin options.

25.3 LEDs and Pulse Width Modulation (PWM)

The device's GPIO pins can also be used for driving external LEDs or any other PWM controlled devices. When the following registers are set and the corresponding bit in GPIO Data Out Enable is enabled, the GPIO pin blinks.

- GPIO_0_31_Blink Enable Register and
- GPIO_32_59_Blink Enable Register

There are 2 global sets (set A and set B) of configurable blinking parameters:

- ON duration, in terms of core clock cycles
- OFF duration, in terms of core clock cycles

The blinking parameters are configured through:

- GPIO Blink Counter A is ON Duration Register
- GPIO Blink Counter A is OFF Duration Register
- GPIO Blink Counter B is ON Duration Register
- GPIO Blink Counter B is OFF Duration Register

Each GPIO can be configure to blink according to set A or set B. See:

- GPIO_0_31_Blink_Counter_Select Register and
- GPIO_32_59_Blink_Counter_Select Register

Document Classification: Proprietary Information

25.4 GPIO Interrupts

When configured to act as GPIO inputs, GPIOs can be used for registration of external devices interrupts into the CPU interrupt controller. The device supports both edge and level sensitive external interrupts.

When the external device uses edge sensitive interrupts, each toggle of a GPIO Data In Register bit from 0 to 1 is registered in the GPIO Interrupt Cause register (see below).
If the interrupt is not masked by the GPIO Interrupt Mask register, the interrupt is routed to the CPU Main Interrupt Cause register.

GPIO Data In Registers

- GPIO_0_31_Data In Register (Table 1422 p. 1142)
- GPIO_32_59_Data In Register (Table 1435 p. 1144)
- GPIO_0_31_Data In Polarity Register (Table 1421 p. 1141)
- GPIO_32_59_Data In Polarity Register (Table 1434 p. 1144)

GPIO Interrupt Cause Registers

- GPIO_0_31_Interrupt Cause Register (Table 1423 p. 1142)
- GPIO_32_59_Interrupt Cause Register (Table 1436 p. 1145)

To clear an edge sensitive interrupt, the software must clear the corresponding bit in the relevant GPIO Interrupt Cause register (see above).

If the external device uses level sensitive interrupts, the data is registered in the GPIO Data In register. If not masked by the GPIO_0_31_Interrupt Level Mask Register or GPIO_32_59_Interrupt Level Mask Register, the interrupt is routed to the CPU Main Interrupt Cause register.



Note

To clear a level sensitive interrupt, the software clears the interrupt directly on the external device.

25.5 Data Input Sampling

The GPIO Data In register samples the data sampled on the GPIO input pins. Data sampled into this register is the data sampled on the pins, or the inverted data, depending on the settings of the:

- GPIO_0_31_Data In Polarity Register (Table 1421 p. 1141) and
- GPIO_32_59_Data In Polarity Register (Table 1434 p. 1144).

25.6 Data Output Driving

Each GPIO can act also as an output. Enable driving data on the corresponding GPIO output pin by setting the:

- GPIO_0_31_Data Out Enable Control Register (Table 1419 p. 1141) and
- GPIO_32_59_Data Out Enable Control Register (Table 1432 p. 1144).

The data driven on the pin is the data configured into the:

- GPIO_0_31_DataOut Register (Table 1418 p. 1141) and
- GPIO_32_59_DataOut Register (Table 1431 p. 1144).

Copyright © 2015 Marvell

The combination of the following registers also enables open drain/open source output implementation.

- GPIO_0_31_Data Out Enable Control Register and GPIO_0_31_DataOut Register,
- GPIO_32_59_Data Out Enable Control Register and GPIO_32_59_DataOut Register

  For example, to implement an open drain output:
  - Set the:
    - GPIO_0_31_DataOut Register/
    - GPIO_32_59_DataOut Register and
  - Toggle the output by setting/clearing the corresponding field in the:
    - GPIO_0_31_Data Out Enable Control Register/
    - GPIO_32_59_Data Out Enable Control Register.

25.7 Multi-CPU Support

Note The following information only applies to devices with more than 1 CPU.

In some multi-CPU applications, GPIO interrupts can be dynamically routed to different CPUs. To support this, the device has a separate masking option of GPIO interrupts per each CPU. Each Mask register controls which of the GPIO interrupts will be routed to the corresponding CPU. The software can dynamically change these masking settings, to change GPIO interrupts routing. For details refer to the ARM® Cortex-A9 documentation (see Related Documents on page 20).

Also, GPIO outputs can be controlled by multiple CPUs. For one CPU not to interfere with the operation of another CPU, the device also implements, in addition to the GPIO_0_31_DataOut Register (Table 1418 p. 1141) and the GPIO_0_31_Data Out Enable Control Register (Table 1419 p. 1141), the Set and Clear registers (see Appendix A.20, General Purpose Input/Output (GPIO) Registers, on page 1140).

For example, if CPU0 needs to set GPIO[0] output to 1, it would typically write 0x1 to the GPIO_0_31_Data Out Enable Control Register (Table 1419 p. 1141). If CPU1 needs to set GPIO[1] output to 1, it would typically write 0x2 to GPIO_0_31_Data Out Enable Control Register (Table 1419 p. 1141). These two writes contradict each other (CPU1 write 0x2 overrides CPU0 setting of GPIO[0]). The device provides a simple hardware mechanism to overcome this race.

Instead of writing to the GPIO Data Out register, the CPU0 writes 0x1 to GPIO_0_31_ Data_Out_Set Register. This write causes bit[0] in the GPIO_32_59_DataOut Register to be set, without affecting any other bits (GPIO Data Out Set register works with write 1 to set, write 0 don't care policy).

Document Classification: Proprietary Information

Similarly, CPU1 writes 0x2 to GPIO_0_31_ Data_Out_Set Register will cause setting of GPIO_32_59_DataOut Register bit[1], while not affecting the setting of bit[0].



Note

- The Set/Clear registers are only relevant only if multiple CPUs are required to simultaneously control the GPIO outputs. If this is not the case, use the conventional GPIO_0_31_Data Out Enable Control Register (Table 1419 p. 1141).
- If the GPIO outputs controlled by each CPU reside in a different byte segment of the GPIOs, the race can be avoided by simply using byte accesses to the GPIO Data Out register. For example, CPU0 controls GPIO[0] and CPU1 controls GPIO[8].
- Similar to GPIO Data Out Set/Clear registers, there are also GPIO Data Output Enable Set/Clear registers. There are useful to solve the above race condition when using the GPIOs as open source/open drain outputs.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family High-Performance Single/Dual CPU System on Chip

Functional Specifications – Unrestricted

Part 6: System Functions and Engines

- Section 26, Real-Time Clock (RTC)
- Section 27, Cryptographic Engines and Security Accelerators (CESA)
- Section 28, Power Management Controller
- Section 29, High-Speed Integrated SERDES Interface
- Section 30, XOR Engines
- Section 31, Electronic Fuse (eFuse)

Doc. No. MV-S109094-U0, Rev. A

November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell

26 Real-Time Clock (RTC)

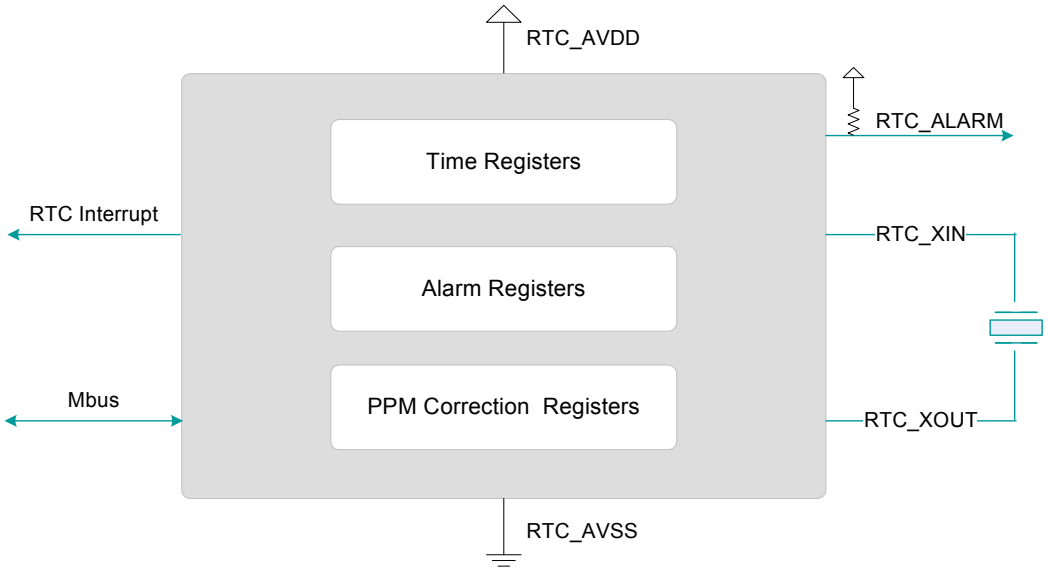
This section describes the device Real-Time Clock (RTC).

The RTC provides an accurate seconds counting circuitry for applications that require low power operation from an external battery with high accuracy. It also provides 2 programmable internal and 1 external alarm for a real-time periodic interrupts.

The external alarm is active during power down modes. It is useful for generating system level alarm events on the device interface during low power modes.

The RTC unit operates with an external 32.768 kHz crystal.

Figure 83: RTC Block Diagram



The RTC registers are located in Appendix A.21, Real-Time Clock (RTC) Registers, on page 1150.

26.1 Features

The RTC features include:

- An accurate 32-bit seconds counter
- 2 internal alarm signaling
- A single external alarm signal driven through an open drain pad
- Low power 32 kHz oscillator
- Clock correction mechanism with an accuracy of up to +/-1 ppm

Copyright © 2015 Marvell

Note

The RTC internal alarms are not powered by a battery power pin. The internal alarm can only be used when the device power is on.

26.2 Status and Control Registers Description

The following sections describes the main RTC registers.

26.2.1 Time Registers

The RTC Time Register (Table 1458 p. 1151) provides the current time value in seconds. See Section 26.3, Seconds to Time Conversion, on page 305 for a description of how to convert free running seconds counting to the Seconds/Minute/Hour/Day/Month/Year format.

The RTC Time Register is programmable.

Note

- A write to the RTC Time Register might have a 32 kHz offset. The host must verify that the time value was captured correctly.
- When the RTC is powered up, the RTC Time Register should be written twice with a 32 kHz period in between to guarantee the write is performed correctly.

26.2.2 Interrupt Status Register

The RTC Interrupt 1 Configuration Register (Table 1456 p. 1150) and RTC Interrupt 2 Configuration Register (Table 1457 p. 1151) provide the current status of the alarm signals.

Note

Clear the RTC Interrupt 1 Configuration Register and RTC Interrupt 2 Configuration Register by a write of 0x1.

26.2.3 Interrupt Control Registers

The RTC Interrupt 1 Configuration Register and RTC Interrupt 2 Configuration Register control the behavior of the RTC interrupts.

The following description applies to the 2 internal alarms and the external alarm interrupts. In this description:

- The term INTx refers to Interrupt 1 (INT1) and Interrupt 2 (INT2).
- The term RTC_ALARMx refers to the interrupts signals generated by the RTC.

The external alarm interrupt is configured using the INT1 registers set.

- The <Int1 Alarm En> field in the RTC Interrupt 1 Configuration Register (Table 1456 p. 1151) and the <Int2 Alarm En> field in the RTC Interrupt 2 Configuration Register (Table 1457 p. 1151) enable the RTC Alarms 1 and 2, respectively, in Time mode. When set, the RTC_ALARMx signal will be asserted when the RTC Alarm 1 Register (Table 1459 p. 1152)/RTC Alarm 2 Register (Table 1460 p. 1152) and the RTC Time Register (Table 1458 p. 1151) values match.
- The <Int1 Freq En> field in the RTC Interrupt 1 Configuration Register (Table 1456 p. 1151) and the <Int2 Freq En> field in the RTC Interrupt 2 Configuration Register (Table 1457 p. 1151) enable the RTC_ALARMx in Frequency mode.
- The <Int1 Freq> field in the RTC Interrupt 1 Configuration Register (Table 1456 p. 1150) and the <Int2 Freq> field in the RTC Interrupt 2 Configuration Register (Table 1457 p. 1151) control the

Copyright © 2015 Marvell

frequency rate of the RTC_ALARMx signal. The configured value sets the toggle frequency in hertz.

For example, if the <Int1 Freq En> or <Int2 Freq En>field is set and the value configured to <Int1 Freq> and <Int2 Freq> is 0xA,the RTC_ALARMx signal will toggle 10 times within 1 second.



Note

The <Int1 Freq En>/<Int2 Freq En> field value has priority over the <Int1 Alarm En> and the <Int2 Alarm En> field value.

26.2.4 Alarm Registers

The RTC Alarm 1 Register (Table 1459 p. 1152) and RTC Alarm 2 Register (Table 1460 p. 1152) configure the time (in seconds) for asserting the RTC_ALARMx signals. When the RTC seconds counter matches the programmed alarm value in the RTC Alarm 1 Register and RTC Alarm 2 Register, the corresponding alarm signal is set for 1 second.

26.2.4.1 RTC Alarm Setting Flow

The following steps describe how a single alarm event is enabled.

1. Configure the <Time> field in the RTC Time Register (Table 1458 p. 1151).
2. Configure the desired event time through the RTC Alarm 1 Register (Table 1459 p. 1152).
3. Enable the relevant RTC ALARM interrupt by writing 0x1 to either <Int1 Alarm En> or <Int2 Alarm En> field.
4. Enable the RTC ALARM interrupt through <RTC ALARM1 Interrupt Mask> field in the RTC SoC Interrupt Register (Table 1911 p. 1411).

Before a new interrupt is enabled, the software must disable the <RTC ALARM1 Interrupt Mask>/<RTC ALARM2 Interrupt Mask> fields and then clear the <RTC ALARM1 Interrupt>/<RTC ALARM2 Interrupt> field in the RTC SoC Interrupt Register (Table 1911 p. 1411).

26.2.4.2 RTC Periodic Alarm Setting Flow

The following steps describe how a periodic alarm event is enabled.

1. Configuring the <Time> field in the RTC Time Register (Table 1458 p. 1151).
2. Configure the desired event frequency through <Int1 Freq> field in the RTC Interrupt 1 Configuration Register (Table 1456 p. 1150).
3. Enable the periodic event by setting either <Int1 Freq En> or <Int2 Freq En>.
4. Enable the RTC ALARM Interrupt through the <RTC ALARM1 Interrupt Mask> field in the RTC SoC Interrupt Register (Table 1911 p. 1411).

26.2.5 Clock Correction Register

The RTC Clock Correction Register (Table 1461 p. 1152) participates in the Clock Correction Implementation flow as defined in Section 26.4, Clock Correction Implementation.

26.3 Seconds to Time Conversion

The RTC implements an accurate 32b seconds counter that represents a range of approximately 136 years.

The conversion from the seconds format to Seconds/Minutes/Hour/Day/Month/Year format should be implemented by software.

Use the following equations for calculating the *Time-now* in seconds based on the Seconds/Minutes/Hour/Day/Month/Year values received from another system.

Copyright © 2015 Marvell

The formula assumes the year 2000 as the initial value.

$$Time\text{-}Now\ in\ seconds = (Year - 2000) * 365 * 86400 + FLOOR((Year - 1997 + (Month>2))/4) * 86400 + DAYS_IN_YEAR(Month) * 86400 + ((Day * 24 + Hours) * 60 + Minutes) * 60 + Seconds$$

FLOOR (value)	Return the result of a round function to the next smaller integer. (Together with the division by 4, this can be simply implemented by a shift right of 2 bits.)
(Month>2)	Return 0 for January and February and 1 for all other months.
DAYS_IN_YEAR	Return the result of a look-up in the following array: [0 31 59 90 120 151 181 212 243 273 304 334]

26.4 Clock Correction Implementation

The Real-Time Clock implements a clock counter correction mechanism to improve the accuracy of the time value. This correction value can be obtained during product assembly and stored in the RTC Clock Correction Register (Table 1461 p. 1152) (if battery power is maintained) or in non-volatile memory that is firmware accessible (if the battery power is not maintained on the RTC). If another source of accurate time measurement is available (for example, an Internet time server) the correction value could further be updated by the firmware.

There are 2 clock correction modes with the specifications listed in Table 72.

Table 72: Clock Correction Mode Specifications

Category	Low Update Mode	High Update Mode
Correction every	32 seconds	8 seconds
Resolution	0.954 ppm	3.815 ppm
Correction range	-122 to 121 ppm	-488 to 484 ppm

To determine which mode can be used calculate the current frequency offset by measuring a 1 Hz clock at 1 of the interrupt outputs (for example, the <Int1 Freq En> field = 1, the <Int1 Freq> field = 1).

$$Offset = (current\ oscillation\ frequency / 32768) - 1$$

If the offset is beyond the range for Low Update Mode, set CMODE (<Correct Mode>) = 1 for High Update mode. Otherwise CMODE should be set to 0 for a better resolution of the correction.

In the formulas below CVALUE corresponds to the value in <Correct Value> field in the RTC Clock Correction Register (Table 1461 p. 1152).

Positive – When the measured offset is positive (clock is fast) the correction value is calculated as follows

$$CVALUE = 256 - ROUND\left(\frac{current_oscillation_frequency - 32768}{current_oscillation_frequency * resolution}\right)$$

Document Classification: Proprietary Information

Negative – When the measured offset is negative (clock is slow) the correction value is calculated as follows:

$$CVALUE = ROUND\left(\frac{32768 - current_oscillation_frequency}{current_oscillation_frequency * resolution}\right)$$

The extension of the correction value has 7 bits and depends only on the MSB of the CVALUE (named C_MSB below). It is calculated as follows (~C_MSB is the inverted C_MSB):

$$CVAL_EXT = \{0, \sim C_MSB, C_MSB, C_MSB, C_MSB, C_MSB, C_MSB\}$$

Copyright © 2015 Marvell

27 Cryptographic Engines and Security Accelerators (CESA)

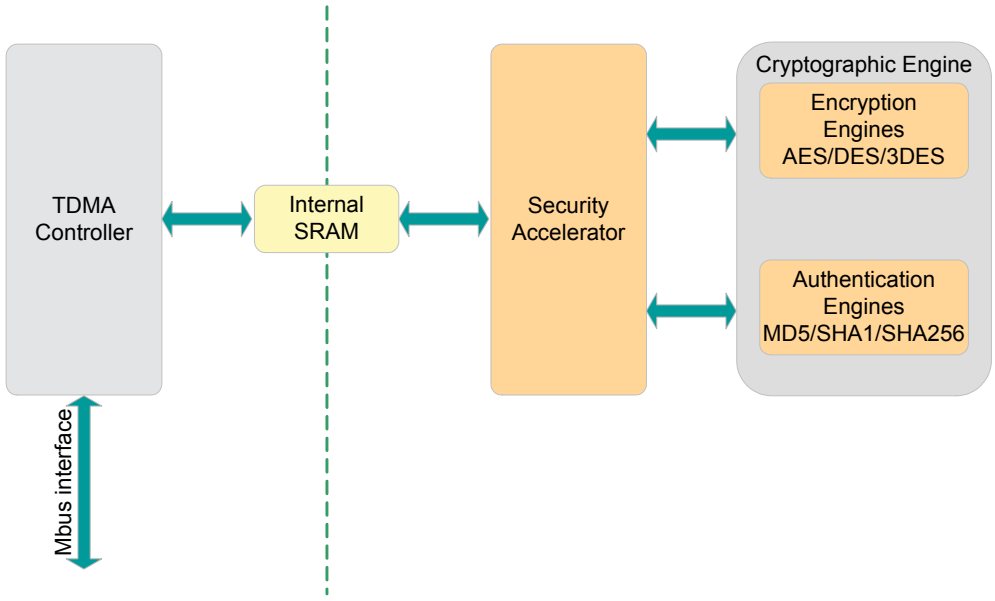
The Cryptographic Engines and Security Accelerator (CESA) reduces the CPU packet processing overhead by performing time consuming cryptographic operations, such as:

- Advanced Encryption Standard (AES)
- Data Encryption Standard (DES)
- Triple Data Encryption Standard (3DES) encryption
- Message Digest 5 (MD5)
- Secure Hash Algorithm-1 (SHA-1)
- Secure Hash Algorithm 2 with 256 digest bit size (SHA-2) authentication

The CESA-DMA engine (also called TDMA) controls communication between the main memory and the internal SRAM.

Figure 84 presents a single CESA unit.

Figure 84: CESA Functional Block Diagram



Acronyms

See Appendix B, Acronyms and Abbreviations, on page 1423, for the definitions of terms used in this section.

Copyright © 2015 Marvell

27.1 Features

The CESA incorporates the cryptographic engines, TDMA engine, and Security Accelerator, as listed in the following sections.

27.1.1 Cryptographic Engine Features

There are 4 independently-operating authentication/encryption engines in a single CESA unit.

- Authentication MD5/SHA-1/SHA-2 engine
- DES encryption/decryption engine
- AES128 encryption engine
- AES128 decryption engine

Each of these four independent cryptographic engines has separate registers for data, control, and operation modes.

These engines implement the following algorithms:

- Encryption: DES (ECB and CBC mode) and Triple DES (ECB and CBC mode, EDE and EEE)
- Encryption: AES128/128, AES128/192, AES128/256.
- Authentication: SHA-1, SHA-2 (with 256-bit digest), and MD5

The cryptographic engines implement the following features:

- Authentication in the MD5, SHA-1, or SHA-2 algorithm, selectable by the user
- Authentication Continue mode, enables chaining between blocks
- Authentication Automatic Padding mode
- Encryption and Decryption in Single DES, Single (ECB) or Block (CBC) mode, or 3DES, EEE or EDE mode, selectable by the user
- DES write pipeline
- Optimal external update of Authentication and Encryption (in CBC) initial values, enabling flexibility of use—multi-packet calculation, sharing between resources
- Byte Swap support for Data input and initial values
- Byte Swap support for DES/3DES and AES data output.
- Automatic engine activation when the required data block is loaded
- Authentication and encryption can be done simultaneously
- Authentication and encryption termination interrupts
- Supports DES OFB and CFB modes with additional software
- AES encryption and decryption—completely separate engines that can work simultaneously when TDMA and the Security Accelerator are not used.

27.1.2 TDMA Engine Features

There is one TDMA engine that implements the following features:

- Moves data back and forth between the main memory and the internal SRAM
- Transfers a single data buffer of up to 2 KB
- Supports Chain mode

27.1.3 Security Accelerator Features

The Security Accelerator utilizes the engines to perform one of the following operations per-packet:

- Performs a complete over-the-packet operation with no software intervention
- Supports 4 types of operation:
  - Authentication only (MD5/SHA-1/SHA-2/HMAC-MD5/HMAC-SHA-1/HMAC-SHA-2)
  - Encryption/Decryption only (DES/3DES/AES - both ECB and CBC)

Copyright © 2015 Marvell

- Authentication followed by Decryption/Encryption
- Decryption/Encryption followed by Authentication

27.2 Theory of Operation

The CESA can work in one of the following operational flows:

- The Software directly controls the cryptographic engine per data block.
- The Security Accelerator performs a complete over-the-packet operation with no software intervention. The software or TDMA copies the packet from/to main memory to/from the internal SRAM.
- An Enhanced Flow that attaches the TDMA to the Security Accelerator on a per packet basis or by a Multi-Packet Chain.

In this flow, the TDMA engine copies packets and descriptors into the internal SRAM, the Security Accelerator controls the packets processing according to the descriptors (using the cryptographic engines), and the TDMA engine copies the processed packet back to the main memory. Multiple packets can be chained and processed by the hardware without software intervention.

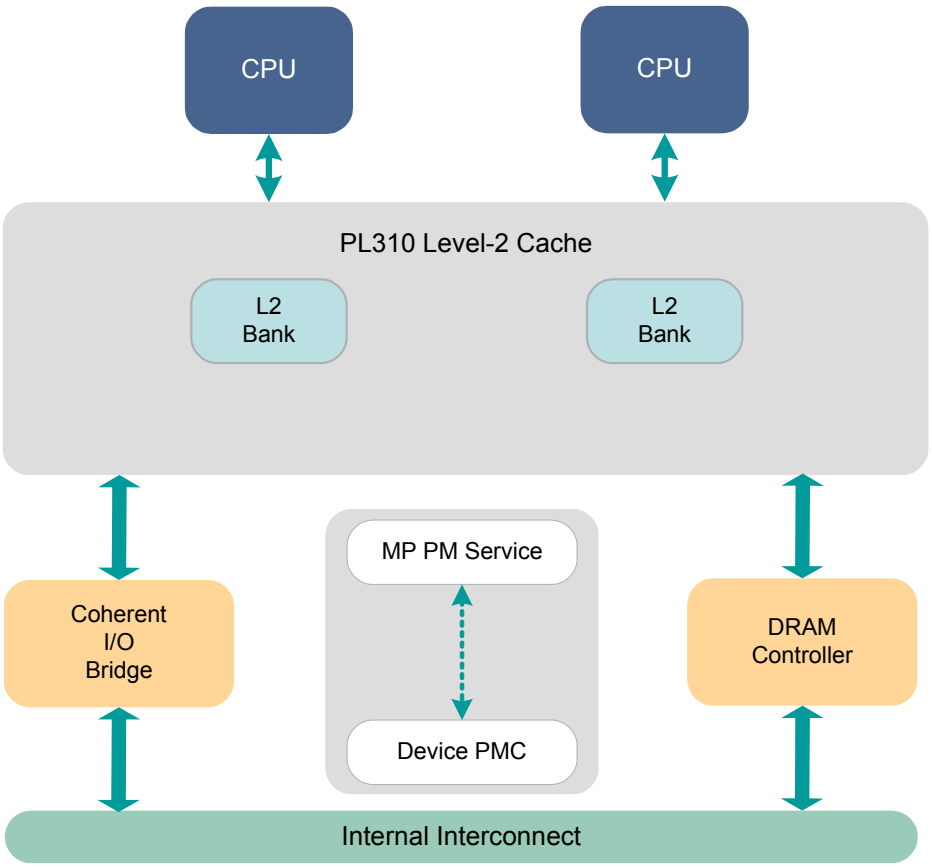
Copyright © 2015 Marvell

28 Power Management Controller

This section describes the power management (PM) capabilities of the device, including programming and signal interfaces.

The Power Management Controller (PMC) administers the power management functions of the device, including reset, clocks, and power domain functional modes. The PMC enables the optimization of the device and system overall power consumption.

Figure 85: Power Management Block Diagram



The Power Management registers are located in Appendix A.3, CPU Power Management Unit Registers, on page 508 and Appendix A.22, Power Management Unit Registers, on page 1153.

28.1 Features

The device supports the following power management features:

- Independent power modes control per CPU with control of on-die, per-CPU power switch for powering down the CPUs
- Run, Idle, and Deep Idle (Power Down) CPU states with coherency management
- Hardware controlled wake-up events and power state transitions
- Dynamic Frequency Scaling (DFS) of all CPU cores, enabling higher performance power flexibility by dynamically changing CPU frequency
- Functional (dynamic or static) clock gating for the CPU subsystem units and the I/O peripherals
- Programmable thermal sensor controller with overheat detection
- Hardware assist for system idle profiling
- Unit specific power management options

28.2 Functional Description

The Power Management functions are provided by 2 power management units:

- The Device Power Management Unit (DEV_PMC)
- The Multiprocessor Power Management Service Unit (MP_PMC).

The MP_PMC is responsible for the power management, clocks, and reset functions of the multiprocessor subsystem, including the CPU cores, the Coherency fabric, the L2 cache and the debug units. The MP_PMC delivers programming interface, wake-up controls and parallel independent entry and exit sequences of each CPU's power modes. In addition, the MP_PMC integrates hardware coordination logic for synchronizing between all the CPU cores for handling shared resources power management flows. The MP_PMC communicates with the DEV_PMC which controls all power domains, external power-supply functions, device PLLs, and peripheral power saving modes.

28.3 Individual CPU Power Management

The device supports the following advanced CPU Core power management capabilities.

Table 73: CPU Core Power Modes

Mode	Description	Core Responds To		
		Snoop	Interrupts	Debug
RUN	All units operating normally with fine-grained dynamic clock gating.	Yes	Yes	Yes
IDLE	CPU is placed in WFI state. It stops dispatching new instructions and halts all internal clocks.	Yes	Yes	Yes
DEEP IDLE (Power Down)	CPU is entirely power-down; caches are flushed to memory.	No	Yes[1]	No

1. While the CPU is in Deep Idle mode, the CPU Power Management unit monitors its unmasked interrupts. In case of an interrupt, the CPU Power Management unit powers up the CPU.

28.3.1 CPU Run Mode

This is a normal mode of operation in which all of the core functionalities are available. The CPU core delivers fine-grained dynamic clock gating to its functional blocks during this mode, lowering the dynamic power consumption during functional operation.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

28.3.2 CPU Idle Mode

Wait For Interrupt (WFI) and Wait For Event (WFE) disables clocks of the unused function, while keeping its logic powered up. Transition to these states is performed directly by software through the execution of a WFI instruction and WFE instructions, respectively. The CPU core ensures that all memory accesses occur in a program order before the WFI operation has completed, and the core enters an idle state. The transition from WFI mode back to Run mode is caused by:

- An interrupt, on the assertion of the IRQ or FIQ
- A debug request, regardless of whether a debug is enabled
- An imprecise Data Abort
- A reset to the CPU

While in WFI/WFE state, the processor serves snoop requests from the coherent fabric. This includes snoops resulting from other CPU cores or I/O agent. Only the blocks that responsible for the snoop processing inside the processor are temporarily woken up and then placed back to WFI/WFE state upon completion of the snoop.

Wait For Event (WFE) instruction is applicable in the multiprocessor. Once a CPU core is being placed in WFE state, it will wake up on any other CPU Send Event notification through execution of the SEV instruction. For example, 2 CPU cores are racing on same semaphore. The one that does not win can insert itself into WFE state, waiting for the other core SEV to release the semaphore.

28.3.3 CPU Deep Idle

Deep Idle state is a low-power state in which power consumption of the CPU core is completely diminished by powering off this domain. There is no processor activity in this mode, state is lost, and the caches contents are not preserved. Thus all processor states, including cache context general purpose registers, system registers in CP15, and debug registers must be saved externally.

The device is brought out of this mode by any enabled a wake-up event. Recovery is fully controlled by the PMC. On exit from Deep Idle low-power mode, the CPU core is resumed through an internal reset.

While in Deep Idle mode, the PMC asserts the appropriate power down signal to the internal power switch.

28.3.3.1 Entering Deep Idle Mode

Before entering Deep Idle mode, the state of the CPU and its caches must be stored to the external SDRAM, including the caches, all ARM registers, all CP15 registers and debug related states.

Enter Deep Idle low-power mode, by performing the following steps:

1. Mask the FIQ and IRQ interrupts in the CPU core by setting the I-bit and F-bit in the CPSR.
2. Flush the L1 Data cache; save all ARM registers, CP15, VFP, and Debug registers.
3. Issue a Data Synchronization Barrier instruction to ensure that all state saving has been completed.
4. Enable any interrupts to be used as wake-ups from low-power Deep Idle mode. If any interrupt is not necessary, it should be disabled through the main interrupt controller IRQ line or at the unit level. Alternatively, in a system where all functional interrupts are mapped to IRQ interrupt, the FIQ can be utilized for triggering a Deep Idle wake-up event by pre-programming all the wake-up interrupts to assert the FIQ. Wake-up triggering through IRQ, FIQ or both are configured by setting the <Enable IRQ Wake Up> and/or <Enable FIQ Wake Up> fields in the ARM® Cortex-A9 CPU registers.
5. To enable debugger wake up during Deep Idle state, set the <Enable Debug Wake Up> field in the ARM® Cortex-A9 CPU registers.

Copyright © 2015 Marvell

6. Set the <Global CPU IRQ Mask> and <Global CPU FIQ Mask> fields in the ARM® Cortex-A9 CPU registers to 1. This masks the main interrupt pins (FIQ and IRQ) from the Interrupt Controller from being asserted to the CPU core.
7. Set the <Deep Idle Power Enable> field in the ARM® Cortex-A9 CPU registers to 1.
8. Place the CPU into Wait For Interrupt mode. When the CPU enters this state, the following steps occur:
   a) CPU activity is halted, outstanding instructions are completed, and the write buffer is drained out.
   b) The CPU clocks are halted.
   c) An internal reset is generated to the CPU.
   d) The PMC powers down the CPU.

28.3.3.2 Behavior During Deep Idle Mode

In Deep Idle mode the processor is powered down and logically placed in reset state. It does not recognize interrupt assertion during Deep Idle mode. Therefore, the MP_PMC monitors pre-programmed wake-up events and invokes exit from Deep Idle mode on any of the device's enabled interrupts. While in Deep Idle state, the <CPU0PwrDwnStts>/<CPU1PwrDwnStts> of the appropriate CPU is set.

28.3.3.3 Exiting Deep Idle Mode

The following occurs after the assertion of any pre-programmed wake-up event:

1. The on-die switch is enabled to resume CPU core voltage. The PMC waits for CPU power to stabilize.
2. The CPU clocks are resumed.
3. The CPU internal reset is de-asserted, enabling CPU fetching from the reset-vector location. All CPU registers are set to their predefined reset exception values.
4. The CPU begins the required hard wired boot sequence that includes the following processes:
   a) Software examines the CPU power status to determine that the reset source was a Deep Idle low-power exit reset.
   b) Software brings the SDRAM out of self-refresh mode, enabling context restoring from SDRAM.
5. The CPU core restores context that was saved before the power down and continues execution.

28.3.4 Dynamic Clock Gating

All the CPU core(s) and co-processor functional blocks integrate cycle accurate dynamic clock gating for halting a non-active functional block, execution component, or pipe stage.

Note

The PMC drives 2 interrupts toward GIC on IDs 105, 106 (CPU0 and CPU1 respectively).

The interrupts represent the aggregation of <CPUn PM Event Status and Mask Register>, n=0-1 fields with their masks:

```
<CPU0DfsDone>   <CPU0DfsDoneMask>
<CPU0PoweredUp>   <CPU0PoweredUpMask>
```


Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

28.4 Dynamic Frequency Scaling (DFS)

The device allows changing of all CPU cores together, to a new target frequency in a specified flow triggered by software.

28.4.1 CPU DFS Flow

The CPU maximum speed is set according to the wake-up strapping configuration as specified by the device Hardware Specifications.

The device supports Dynamic Frequency Scaling of all the CPU cores (together) to half of the maximum frequency through software triggered flow.

To set the new CPU frequency, complete the following steps:

1. Enable GIC.
2. Set interrupts 105 and 106 to be secured.
3. Enable interrupt 105 and 106 at GIC, via the <CPU0Idle0Mask> field in the CPU0 PM Status and Mask Register (Table 271 p. 510) and the <CPU0DfsDoneMask> field in the CPU0 PM Event Status and Mask Register (Table 276 p. 511).
4. One of the CPUs will make the first request:
   a) Configure <cpu_clkdiv_reload_smooth> field in the CPU PLL Clock Divider Control 0 Register (Table 1870 p. 1386).
   b) Configure <cpupll_clkdiv_reset_mask> field in the CPU PLL Clock Divider Control 1 Register (Table 1871 p. 1387).
   c) Set <Cpu Cpu0 ClkDiv Ratio Full> field in the CPU Divider Clock Control Ratio Register (Table 1884 p. 1392).
   d) Set <PmuDfsOvrdEn> field in the PMU Activate I/F Control Register (Table 1468 p. 1154).
5. Program the target frequency PLL ratios, using the <DfsRatio1> field in the PMU DFS Control 1 Register (Table 1467 p. 1154).

   Set the <DfsRatio1> field based on the <DfsInitRatio1> field in the PMU DFS Control 1 Register (Table 1467 p. 1154) setting.

   - Set <DfsRatio0> to 1:2.
6. Enable the CPU DFS Done Interrupt.
7. Set <CPU0DfsReq>/<CPU1DfsReq>.
8. Place the CPU in the Wait For Interrupt mode. When the CPU enters this state, the following steps occur:
   a) CPU activity is halted, outstanding instructions are completed, and the write buffer is drained.
   b) The CPU clocks are halted.
   c) PMU reloads new clocks ratio and set <CPU0DfsDone>/<CPU1DfsDone> on completion; a completion interrupt is asserted to CPU.

To use a single/dual CPU DFS:

- For a single CPU: Set CPU1's <CPU1FrequencyID> field in the CPU1 PM Control and Configuration Register (Table 280 p. 513).
- For a dual CPU: For CPU0, complete the flow above. For CPU1, enable its interrupt and go into WFI while synced with CPU0.

Copyright © 2015 Marvell

28.5 Waking Options

The device provides several waking options sourced by different peripherals to take the device out of power save modes:

- Wake on LAN (wake up by specific packets coming from the Ethernet)
- Wake on USB
- Wake on Timer
- Wake on GPIO (an external event, such as an RTC Alarm, or an interactive event, such as someone touching a touch screen)
- Wake on any other interrupt source

When the system is put in low power mode, it characterizes what events should wake it up and what traffic should be ignored and dropped. Notice that the same traffic may be accepted in normal operational mode, and ignored in low power state. The purpose is only to wake up the system on events that follow the system characterization as awaking events.

28.5.1 Wake On LAN (WOL) Mode

The WOL mode is an option for the system to wake up based on specific packets that enter the Ethernet port.

The Ethernet MAC provides the ability to filter out any packet that does not follow-up the waking packet definition. The Ethernet MAC can also accept the waking packet and wake up the system as a result of this packet.

Once set to WOL mode, the MAC discards all packets that do not pass the WOL filtering.

28.5.2 Wake On USB

The device provides the option to wake up from 1 of the system power save modes by an event triggered on the USB interface.

When operated as a USB peripheral, the device may be put in suspend mode by the remote host. Suspend is signaled by 3 ms of idle time on the bus. When the remote host signals a resume or reset command on the bus, the USB peripheral unit takes the device out of suspend mode and will set a maskable interrupt when the link is operational to wake up the CPU.

When operating as a USB host and before entering CPU power down mode, the software may choose to put the USB port into low power state without disconnecting the link by signaling the connected devices to go into suspend mode.

Once the link is in suspend mode and CPU is powered down, there are several events that may restore the link:

- Remote resume request by a connected USB peripheral
- Connection of a new device
- Disconnection of a connected device

The USB host controller provides the option to disable waking up by any of the events through a software configuration setting. While sensing any of the enabled events, the USB host controller will initiate a maskable interrupt to wake up the CPU once there is a change in the link status.

Shutting down the PHY forces a link disconnect.

28.5.3 Wake on Timer

Before going into power down mode, the software may set one of the device's timers to wake up the system after a preset time period. Once timer is expired, an interrupt will assert triggering a wake up event.

Copyright © 2015 Marvell

28.5.4 Wake on GPIO

External events are one of the most common way to wake up a powered down system, for example, touching a touch screen in a printer, a button that is pushed, a connected device that is connected over one of the powered down interfaces (e.g PCIe) and requests service and so on. When the system is powered down, any Interrupt enabled GPIO assertion will be treated as a waking event to the system.

28.5.5 Other Waking Events

Other system events that may be used as waking alerts are Mbus to DRAM requests for a specific master ID (instead of using the interrupt for reducing the wake latency). Any other system event that triggers an unmasked interrupt cause also wakes the CPU. This includes. for example, SD card insertion, doorbell interrupt in a multi-CPU environment, and MSIx message.

28.6 Thermal Management

The device incorporates a Thermal Management engine for monitoring die temperature. It includes an on-die analog-to-digital thermal sensor, that is used to determine when the maximum specified processor junction temperature has been reached.

Thermal Management functionality is controlled by the:

- Temperature Sensor 28nm Control LSB Register (Table 1875 p. 1388)
- Temperature Sensor 28nm Control MSB Register (Table 1876 p. 1389)

The Temperature Sensor Status Register (Table 1877 p. 1390) provides the thermal status.

The Thermal Manager is disabled by setting <Tsen Power Down> field in the Temperature Sensor 28nm Control MSB Register (Table 1876 p. 1389).

The current measured on-die temperature is a digital value latched to the <Temperature Sensor Readout> field in the Temperature Sensor Status Register (Table 1877 p. 1390) for reading.

The Thermal sensor is active in all power modes.

28.7 Idle Profiling Counter

The profiling counter is activated by enabling the statistics counter. The profiling counter provides the number of idle indications for each CPU within a certain timeframe. The timeframe during which the counter will be active is set using the CPUn Idle Timer Value register. At the end of the timeframe, the value of the counter will be compared to the low threshold and high threshold set by the CPUn Idle High Threshold register and CPUn Idle Low Threshold register. If the value is lower than the low threshold or higher than the high threshold, the value will be locked in the CPUn Idle Result register and a the corresponding interrupt will be generated. Otherwise, the CPUn Idle Result register will contain a value of 0, and the counter will continue to count.

Configuration

- To enable the counters, set the CPUn Statistic Counter Global Control register.
- To configure the timeframe during which the counter will be active, set the CPUn Idle Timer Value register.
- To configure the high threshold for the idle counter, set the CPUn Idle High Threshold register.
- To configure the low threshold for the idle counter, set the CPUn Idle Low Threshold register.
- The value of the idle counter, in case it is greater than the high threshold or smaller than the low threshold (0 otherwise), is indicated by the CPUn Idle Result register.
- To configure the masks for the related idle counter interrupts and the status of each interrupt, set the CPUn PM Event Status and Mask register.

Copyright © 2015 Marvell

28.8 Unit Specific Power Management Options

The device supports the following power saving options in addition to and independent from the CPU's power save options:

- DRAM power save modes – Self refresh, active power down mode and precharged power down mode.
- All PCIe Device, Link, and ASPM Power Modes.
- SERDES power down—Each SERDES may be shut down independently for zero current drawing.
- USB PHY can be either shutdown completely for zero current drawing or put in SUSPEND mode for auxiliary current drawing.
- Each unit has the option to gate its clocks and completely halt to save dynamic power.
- EEE (Energy Efficient Ethernet) – LPI mode. The capability to put the Ethernet PHY into a low power state and resume without breaking the link.

28.9 Debug During Power Management (CoreSight)

During Deep-IDLE state, the CPU Core's debug and tracing logic is powered down as well. Any debug or tracing attempt results in slave error indication from the CoreSight DAP controller.

28.10 Power Management Registers

Table 74 lists the register fields used for the CPU subsystem power management.

Table 74: CPU Core Power Management Register Pointers

Register Field Function	Register Mapping
CPU DFS Request	The <CPUnDfsReq> field (where n=0–1) in: • CPU0 PM Control and Configuration Register (Table 270 p. 509) • CPU1 PM Control and Configuration Register (Table 280 p. 513)
CPU DFS Done	The <CPUnDfsDone> field (where n=0–1) in: • CPU0 PM Event Status and Mask Register (Table 276 p. 511) • CPU1 PM Event Status and Mask Register (Table 286 p. 515)
Global CPU IRQ Mask	The <CPUnIntr0Mask> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)
Global CPU FIQ Mask	The <CPUnIntr1Mask> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)
Enable IRQ Wake Up	The <CPUnWake0Mask> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)
Enable FIQ Wake Up	The <CPUnWake1Mask> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)
Enable Debug Wake Up	The <CPUnWake2Mask> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)

Copyright © 2015 Marvell

Table 74: CPU Core Power Management Register Pointers (Continued)

Register Field Function	Register Mapping
CPU Idle Wait Mask	The <CPUnIdle0Mask> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)
Deep Idle Power Enable	The <CPUnPowerDownReq> field (where n=0–1) in: • CPU0 PM Control and Configuration Register (Table 270 p. 509) • CPU0 PM Control and Configuration Register (Table 270 p. 509)
CPU Power Down Status	The <CPUnPwrDwnStts> field (where n=0–1) in: • CPU0 PM Status and Mask Register (Table 271 p. 509) • CPU1 PM Status and Mask Register (Table 281 p. 513)

29 High-Speed Integrated SERDES Interface

This device integrates Marvell high-speed Serializer/Deserializer (SERDES) interfaces.
The SERDES lanes may be used as the electrical interface for various functionalities, such as:

- SATA 3.0/2.0
- PCI Express 2.0/1.1
- USB 3.0
- SGMII
- QSGMII

Note The SATA and QSGMII interfaces are only supported by 88F6925 and 88F6928.

This versatility provides a large degree of flexibility for different types of applications.

A set of configuration registers is used to assign a SERDES lane to a specific operation mode. Configuring these registers must be completed before the link on the SERDES is established. The following steps are required to create the SERDES allocation:

- Create the SERDES allocation on the device pins via Common PHYs Selectors Register (Table 1479 p. 1168). Each field in this register represents a corresponding SERDES lane. The field value determines if the SERDES is active or not and sets the functionality mode of the SERDES.
- Configure the relevant MAC and SERDES to the specific work mode as described in the following sections.

Note A full matrix of configuration options for this device is included in the *High-Speed SERDES Multiplexing* section and the *SERDES Lane Selection Options* table in the device Hardware Specifications.

29.1 Features

The high-speed integrated SERDES interface supports:

- Multiple integrated low-power, high-speed SERDES lanes, based on Marvell® SERDES technology
- Versatile multiplexing options of SATA, PCIe, USB 3.0, SGMII, and QSGMII interfaces

29.2 SERDES Configuration

The device implements a management of the SERDES as follows:

- SERDES Internal Register File Access
- SERDES External Configuration Control
- SERDES External Statuses Visibility

Document Classification: Proprietary Information

29.2.1 SERDES Internal Register File Access

Using an internal configuration type transaction, the host can directly access the SERDES internal register file for both writes and reads.

See the Appendix A.24, Common PHY and PIPE Interface Registers, on page 1170.

29.2.2 SERDES External Configuration Control

The Common PHY <n> Configuration <x> registers (Table 1470–Table 1474) provide a way for controlling the SERDES external configurations.

29.2.3 SERDES External Statuses Visibility

The Common PHY <n> Status <x> registers (Table 1475–Table 1478) provide a way for observing the SERDES external statuses.

29.2.4 Configuration Exceptions

The following configurations are not driven from the Common PHY <n> Configuration 1–6 registers (Table 1470–Table 1474):

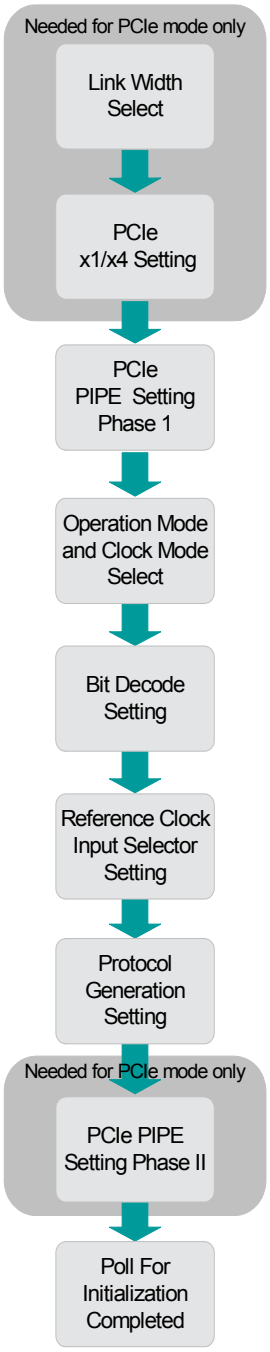
- **SATA:**
  When configuring a COMPHY selector to work with a SATA MAC, the following configurations are driven from the SATA MAC, controlled from the SATA MAC internal register file, instead of the Common PHY <n> Configuration 1–6 registers.
  - GEN TX <Gen TX> in Table 1470
  - GEN RX<Gen RX> in Table 1470
  - Power Up TX<Power Up TX> in Table 1470
  - Power Up RX<Power Up RX> in Table 1470
  - Power Up PLL<Power Up PLL> in Table 1470
  - RX INIT<RX INIT> in Table 1470
  - Pin TX Idle<Pin TX Idle> in Table 1470
- **PCIe and USB 3.0:**
  When configuring a COMPHY selector to work with a PCIe or USB 3.0 MAC, the following configurations are driven from the MAC internal register file instead of the Common PHY <n> Configuration 1–6 registers.
  - <TXDEEMPH> in Table 1470
  - <MacPhyTxElecIdle> in Table 1470
  - <MacPhyPwDn0> in Table 1471
  - <MacPhyRxPolarity> in Table 1471
  - <MacPhyTxDetectRxLoopBack> in Table 1471
  - <MacPhyTxMargin> in Table 1471
  - <MacPhyTxSwing> in Table 1471

Copyright © 2015 Marvell

29.3 Assigning a SERDES Lane to the Relevant MAC

After selecting the operation mode for a given SERDES via registers Common PHYs Selectors Register (Table 1479 p. 1168), the software must configure the relevant MAC and SERDES operation modes. Figure 86 outlines the steps required to complete the MAC and SERDES configuration before the link may be used for an operational mode.

Figure 86: Assigning SERDES to a MAC



Copyright © 2015 Marvell

The following sections describe the exact configurations required for each of these steps.

29.3.1 PCIe Link Width Select

Some of the PCIe units have the option of working in either x4 link width or quad x1 independent links. Set the <PCIe x4 Enable> field in the Common PHYs Selectors Register (Table 1479 p. 1168) to the required operational mode (x4 or quad x1).


Note

Not all PCIe units have the option to be configured as x4 or quad x1. A unit may have a single option—either x4 or x1. Refer to Section 17, PCI Express Interface (PCIe) 2.0, on page 207 for more details on the device's PCIe unit setting options.

In addition, configure the width of physical interface (x4 or x1) in each PCIe controller via the relevant controller.


Note

Only values of x4 or x1 are supported. Other values are reserved.

29.3.2 PCIe x4 Setting


Note

The configuration below are additional configuration for PCIe x4 mode, and should be implemented in addition to the regular (x1) PCIe link settings.

To establish PCIe x4 links, configure the following:

1. Set the Common PHYs selectors to the PCIe x4 controller, by configuring the following fields in the Common PHYs Selectors Register (Table 1479 p. 1168):
   - <COMPHY0 Selector> = 0x1.
   - <COMPHY1 Selector> = 0x2.
   - <COMPHY2 Selector> = 0x2.
   - <COMPHY3 Selector> = 0x2.
2. Set the 4 Common PHYs for x4 mode by setting the <PCIe x4 Enable> field in the Common PHYs Selectors Register (Table 1479 p. 1168).
3. Set the PCIe controller for x4 mode by setting the <PCIe0QuadBy4En> field in the SoC Control 1 Register (Table 1888 p. 1395).

29.3.3 PCIe PIPE Setting Phase I


Note

These settings are only required if the PCIe is used.

The settings in Table 75 are needed to configure the SERDES to work in PCIe mode. Each row in the table defines an access to one of the SERDES registers.

Table 75: PCIe PIPE Configurations

SERDES Register	Data	Operation
Reset and Clock Control Register (n=0–1) (Table 1799 p. 1343)	0x25	Using the Reset and Clock Control Register (n=0–1) (Table 1799 p. 1343): Set soft reset by configuring the <PIPE_SFT_RESET> field to 1.
Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)	0xfc60	Using the Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)[1]: 1. Set PHY to PCIe mode, by configuring the <PHY_MODE[2:0]> field to 3. 2. Set Reference clock frequency to 100 MHz, by configuring the <REF_FREF_SEL[4:0]> field to 0.
Sync Mask and Maximal Generation Setting Register (n=0–6) (Table 1518 p. 1233)	0x17ff	Set the maximum supported data rate to 5 Gbps by configuring the <PHY_GEN_MAX[1:0]> field in the Sync Mask and Maximal Generation Setting Register (n=0–6) (Table 1518 p. 1234) to 1.
Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)	0xa08a	Using the Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254): 1. Set the <TXDCLK_2X_SEL> field to 1. Pin TXDCLK_2X sends out a 500-MHz clock. 2. Set the <CLK500M_EN> field to 1. This enables the 500-MHz clock. 3. Set the <REFCLK_SEL> field to 0. This set the reference clock to come from group 1.
Reset and Clock Control Register (n=0–1) (Table 1799 p. 1343)	0x24	Release non-register soft reset by configuring the <PIPE_SFT_RESET> field to 0. **NOTE:** There should be no register programming after this point.

1. For all references to this register, in this device only n = 0–5 applies.

29.3.4 Operational Mode and Reference Clock Setting

To determine the operational mode of the SERDES and the reference clock frequency of the internal SERDES PLL, configure the <PHY_MODE[2:0]> field and the <REF_FREF_SEL[4:0]> field in the Power and PLL Control Register (n=0–6) (Table 1482 p. 1207) of the corresponding SERDES.

Table 76 lists the required setting per each operational mode and PHY selection.

Table 76: PHY MODE and REF_FREF_SEL Setting

Interface	<PHY_MODE [2:0]>field	<REF_FREF_SEL [4:0]> field	Register
PCIe	0x3 (PCIe)	0x0 (100 MHz)	Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)[1]
SATA	0x0 (SATA)	0x1 (25 MHz)	Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)
USB 3.0	0x5 (USB)	0x2 (25 MHz)	Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)
SGMII	0x4 (SERDES)	0x1 (25 MHz)	Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)
QSGMII	0x4 (SERDES)	0x1 (25 MHz)	Power and PLL Control Register (n=0–6) (Table 1482 p. 1205)—Configure Port 0

1. For all references to this register, in this device only n = 0–5 applies.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

29.3.5 Bits Decode Setting

To determine the bits decode of the SERDES to be either 10 or 20-bits according to the required link speed, configure the <SEL_BITS[2:0]> field in the Digital Loopback Enable Register (n=0–6) (Table 1516 p. 1232) of the corresponding PHY. As each PHY contains a different Digital Loopback Enable register, only apply the setting to the relevant PHY as selected by the SERDES multiplex options. For example, if chosen to work with PCIe lane 0.x, configure the Digital Loopback Enable Register (n=0–6) (Table 1516 p. 1231) of PCIe PHY 0 only.

Table 77 lists the required setting per each operational mode and MAC selection.

Table 77: SEL_BITS Settings

Interface	SEL_BITS field	Register
PCIe	NA	NA Done automatically by the SERDES while in PCIe mode.
SATA	0x2 (40bits)	Digital Loopback Enable Register (n=0–6) (Table 1516 p. 1231)
USB 3.0	NA	NA Done automatically by the SERDES while in USB 3.0 mode.
SGMII	0x1 (20bits)	Digital Loopback Enable Register (n=0–6) (Table 1516 p. 1231)
QSGMII	0x1 (20bits)	Digital Loopback Enable Register (n=0–6) (Table 1516 p. 1231)—Configure Port 0

29.3.6 Reference Clock Select Setting

To determine the source clock input to the reference clock of the integrated SERDES PLL, configure the <REFCLK_SEL> field in the Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)[1] of the corresponding PHY. As each PHY contain a different Miscellaneous Control 0 register, only apply the setting to the relevant PHY as selected by the SERDES multiplex options. For example, if chosen to work with PCIe lane 0.x, configure Miscellaneous Control 0 register of PCIe PHY 0 only.

Table 78 outlines the required setting per each operational mode and MAC selection.

Table 78: REFCLK_SEL Setting

Interface	REFCLK_SEL field	Register
PCIe	0x0	Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)[1]
SATA	0x1	Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)
USB 3.0	0x1	Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)
SGMII	0x1	Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)
QSGMII	0x1	Miscellaneous Control 0 Register (n=0–6) (Table 1560 p. 1254)—Configure Port 0.

1. For all references to this register, in this device only n = 0–5 applies.

1. For all references to this register, in this device only n = 0–5 applies.

29.3.7 Protocol Generation Setting

To determine the protocol generation in which the interface will work, configure the Common PHY<n> Configuration 1 Register (n=0–5) (Table 1470 p. 1156) for the SATA ports and for the SGMII/QSGMII ports of the corresponding PHY. As each PHY contains different such registers, only apply the setting to the relevant PHY, as selected by the SERDES multiplex options. For example, if you have chosen to work with SGMII 0, configure the register of SGMII MAC 0 only.

Table 79 lists the required settings per each operational mode and MAC selection.

Table 79: PIN_PHY_GEN Setting

Interface	PIN_PHY_GEN field	Register
PCIe	NA	PCIe standard defines specific flow to determine the link generation and speed. Refer to the PCIe section for more details.
SATA	Gen III (6 Gbps) 0x11 Gen II (3 Gbps) 0x10 Gen I (1.5 Gbps) 0x00	<Speed SCR2> field in the SATA Control (SControl/SCR2) Register (i=0–1, n=0–1) (Table 1286 p. 1049)
USB 3.0	NA	USB 3.0 standard defines specific flow to determine the link generation and speed. Refer to the USB 3.0 section for more details.
SGMII	SGMII (1.25 Gbps) 0x6 HS-SGMII (3.125 Gbps) 0x8	Common PHY<n> Configuration 1 Register (n=0–5) (Table 1470 p. 1156)
QSGMII	QSGMII (5 Gbps) 0x3	Common PHY<n> Configuration 1 Register (n=0–5) (Table 1470 p. 1156)— Configure Port 0

29.3.8 PCIe PIPE Setting Phase II

Note These settings are only required if the PCIe is used.

Table 80 lists the settings that are required to configure the SERDES to operate in PCIe mode.

Table 80: PCIe Pipe Configurations

SERDES Register	Operation	Address	Value
Reset and Clock Control Register (n=0–1) (Table 1799 p. 1343)	Remove PIPE soft reset + fixed pclk mode	0x000A0704	0x24

Copyright © 2015 Marvell

29.3.9 Poll for Initialization Completed

To complete the configuration process, the CPU must poll the completion indications from the various SERDES lanes, as listed in Table 81. For SERDES lanes configured for PCIe, the PCIe standard defines specific flow to determine the link status.

Table 81: Completion Indications

Interface	Completion Value (CPU should poll for this value)	Register (Completion status will appear in this register)
SERDES0	Bits[3:2] = 0x3 and bit[0] = 0x1	Common PHY<n> Status 1 Register (n=0–5) (Table 1475 p. 1164) **NOTE:** n = 0–5 corresponds to SERDES ports 0–5.
SERDES1	Bits[3:2] = 0x3 and bit[0] = 0x1	
SERDES2	Bits[3:2] = 0x3 and bit[0] = 0x1	
SERDES3	Bits[3:2] = 0x3 and bit[0] = 0x1	
SERDES4	Bits[3:2] = 0x3 and bit[0] = 0x1	
SERDES5	Bits[3:2] = 0x3 and bit[0] = 0x1	

Once a completion indication has been received, the SERDES and MAC are programmed to work in the desired operation mode.



Note

Completing this stage connects a specific MAC to a SERDES and configures the operational mode of the link on the specific SERDES lane.

To establish the external interface link, the software must complete link establishment, as required by the specific interface protocol specifications.

29.4 Configuring Amplitude and Pre-emphasize Level

To select the amplitude/pre-emphasize—AMP/EMPH/Slew level—in PCIe Gen1/Gen2 modes:

- For Amplitude configure PCIe Gen1/Gen2 modes – Configure the <tx_amp_pipe_v0[4:0]> field in the TX Amplitude Setting for PIPE Register (n=0–6) (Table 1639 p. 1283).
- For Pre-emphasize PCIe Gen1 mode and for PCIe Gen2 mode is configured to 3.5 dB – After configuring the MAC to 3.5 dB mode, configure the <tx_emph_pipe1[3:0]> field in the TX Margin Selection Register (n=0–6) (Table 1640 p. 1283).
  For fine-tuning the pre-emphasis rate in 3.5 dB mode, use <ComplianceDeemphasis> field in the PCI Express Link Control Status 2 Register (p=1–3) (Table 479 p. 654).
- For Pre-emphasize PCIe Gen2 mode to be configured to 6 dB – After configuring the MAC to 3.5db mode, configure the <tx_emph_pipe0[3:0]> field in the TX Margin Selection Register (n=0–6) (Table 1640 p. 1283).
- For Slew rate control in PCIe Gen1 mode – Configure the <G1_TX_SLEW_RATE_SEL[2:0]> field in the Generation 1 Setting 0 Register (n=0–6) (Table 1494 p. 1216).
- For Slew rate control in PCIe Gen2 mode – Configure the <G2_TX_SLEW_RATE_SEL[2:0]> field in the Generation 2 Setting 0 Register (n=0–6) (Table 1496 p. 1218).

Document Classification: Proprietary Information

30 XOR Engines

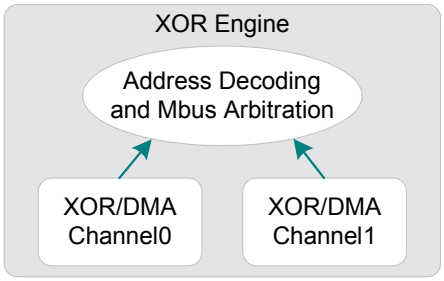
This device integrates 2 dual-channel XOR engines (overall 4 XOR DMA channels).
This section describes a single channel XOR engine, but applies to all other channels/engines as well.

The XOR engine is a generic acceleration engine for storage applications that provides a low-latency, high-throughput XOR calculation capabilities, enabling CPU XOR calculation offloading in various RAID implementations, as well as RAID6 calculation. In addition, the engine provides DMA (Memory copy) functionality, iSCSI CRC32C calculation, memory ECC errors cleanup operation and memory initialization (Memory set, Memory clear) support.

The XOR engine enables to offload or accelerate the CPU and speeds up overall system performance in Networking and Storage applications.

Figure 87 presents an XOR engine design with 2 channels.

Figure 87: XOR Engine Block Diagram



The XOR engine registers are located in Appendix A.25, XOR Engine Registers, on page 1363.

30.1 Features

- Two separate channels for enabling concurrent operation (for example, concurrent XOR and iSCSI CRC32C calculations)
- 512-byte buffer per channel
- Support unaligned data transfers
- XOR calculation for up to eight data block sources
- RAID6 calculation using the $x^8+x^4+x^3+x^2+1$ polynomial.
- RAID6 P and Q generation
- RAID6 reconstruction for up to 2 corrupt buffers (either data, P or Q)
- Multiplication over Galois Field function to enable RAID6 for more than 8 source buffers
- Data block size up to 16 MB
- Programmable maximum burst size on read and write
- Descriptor chain mechanism
- Hot insertion of new descriptors to chain
- iSCSI CRC32C calculation that is compliant with IPS iSCSI version 13 draft
- DMA operation

Copyright © 2015 Marvell

- Memory initialization support
- Memory ECC cleanup support
- Write access protection of configuration registers

30.2 Functional Description

The XOR engine has 6 main operation modes:

- XOR calculation Mode (XOR)
- RAID6 Generation and Recovery Modes
- iSCSI CRC32C Calculation Mode (CRC)
- DMA Operation Mode (DMA)
- Memory initialization Mode (MemInit)
- Memory ECC error cleanup mode (ECC)

The engine has 2 independent channels. Each channel can be configured in 2 ways.

- One way is to have each channel perform a single operation mode at a time. The operation mode is defined through the <OperationMode> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1372). In this mode the XOR engine can perform the following operations: XOR, CRC and DMA.
- The other way is to have the mode of operation encoded in each of the descriptors in the chain. this way each chain of descriptors can be used to perform different operations, not limiting the use of one operation mode per one channel. The operation mode is defined when the <OperationMode> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1372) is set to *Descriptor*. In this mode the XOR engine can perform the following operations: XOR, CRC, DMA, RAID6-P and Q Generation, RAID6 Recovery, and RAID6 Multiply.

For XOR, CRC, DMA, RAID6-P and Q Generation, RAID6 Recovery and RAID6 Multiply, the XOR engine is controlled by chain of descriptors and responds to similar activation scheme. These modes differ only in the interpretation of the chain descriptor fields. In ECC and Memory Initialization (MemInit) modes, the XOR engine responds to different activation schemes. It is controlled by programming internal registers directly. On all operation modes, XOR engine uses the same address decoding scheme.

Upon startup, the 2 XOR channels are in an inactive state and can be configured to any operation mode (XOR, CRC, DMA, ECC, MemInit or Descriptor). After being configured, the XOR engine channel can be activated. It can be stopped or paused by software at any time. After stopped by software, the engine re-enters inactive state and can be configured to another operation mode and re-activated. This also applies if the XOR engine channel finished the operation (reached End Of Chain) without being stopped by the software. Again, the engine re-enters an inactive state and can be configured to another operation mode, and re-activated. After paused by software, the XOR engine channel suspends the current operation at the earliest opportunity. Upon activating the channel again, it resumes executing the same operation.

The two XOR engine channels are independent in their operation modes. The only exception is that both engines must not be configured to ECC or MemInit operation modes. These modes share hardware resources.



Note

Attempting to change the channels operation mode during a pause results in unexpected behavior.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

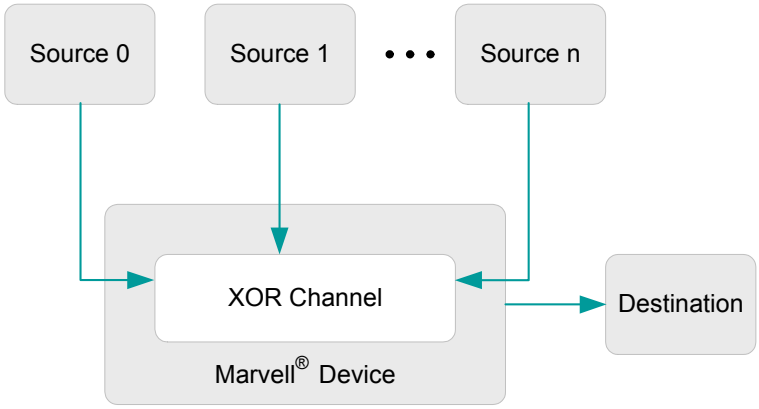
30.3 XOR Engine Modes of Operation

30.3.1 XOR Engine Operation

The XOR engine enables block XOR calculation in hardware. It performs the XOR operation on multiple blocks of source (incoming) data and stores the result back in a destination block. The source and destination addresses are specified through a chain descriptor.

Figure 88 shows how the XOR operation works with multiple blocks of source (incoming) data and stores the result back in a destination block.

Figure 88: XOR Operation with Multiple Incoming Data Blocks



The parameters of the XOR operation are configured by writing the relevant information to a chain descriptor. The relevant parameters consist of source addresses, destination addresses, the number of bytes to transfer, and various control information.

When activated in XOR mode, the XOR engine fetches the first descriptor and starts performing the XOR operation according to its parameters. After finishing the operation, the XOR engine closes the descriptor by writing back the status word to the descriptor and returning the ownership of it to the CPU. The XOR engine checks whether it reached the end of the descriptor chain. If it is the end, the engine enters an inactive state and waits to be re-activated by software. If it did not reach the end of the chain, it progresses to the next descriptor, and so on.

The basic XOR operation algorithm is as follows:

1. Read data from the first enabled source block to the internal buffer.
2. Read from the second enabled source block and calculate XOR with the data from the internal buffer. The intermediate result is stored in the internal buffer.
3. Step 2 is repeated for the rest of the source buffers until all enabled source buffers are handled (up to 8 sources).
4. Write the internal buffer to the destination buffer.
5. Repeat stages 1-4 until the descriptor byte count in is reached.

30.3.2 RAID6

The XOR engine supports RAID6 both for generation and reconstruction of data buffers.

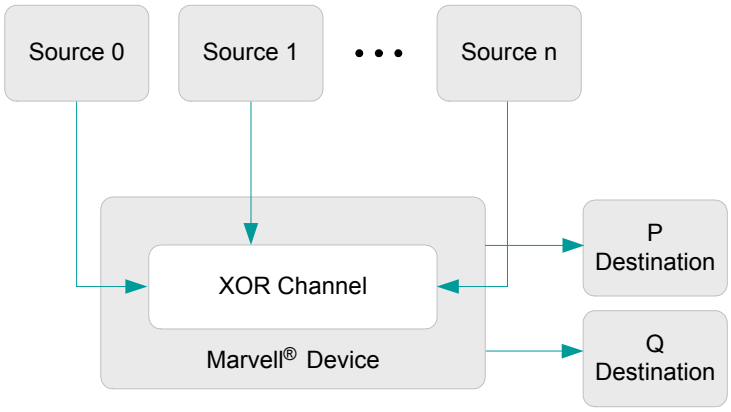
The XOR engine calculates RAID6 P buffer using normal XOR operation and uses the following polynomial for performing calculation of the RAID6 Q buffer: $X^8+X^4+X^3+X^2+1$.

30.3.2.1 P and Q Generation Mode

In P and Q Generation mode, the XOR engine reads source data from the source buffers (supports 1–8 source buffers), and can either calculate P and Q simultaneously, or calculate only one of them, and output the results to dedicated P and Q destination buffers. Using both P and Q calculation simultaneously results in better performance, as source data is read once.

Figure 89 shows the flow of the RAID6 generation operation using *n* input buffers.

Figure 89: RAID6 Generation Flow Diagram



The parameters of the RAID6 P and Q Generation operation are configured by writing the relevant information to a chain descriptor. The relevant parameters consist of source addresses, destination addresses, the number of bytes to operate on, and various control information.

When activated in RAID6 P and Q Generation mode, the XOR engine fetches the first descriptor and starts performing the RAID6 P and Q Generation operation according to its parameters. After finishing the operation, the XOR engine closes the descriptor by writing back the status word to the descriptor and returning the ownership of it to the CPU. The XOR engine checks whether it reached the end of the descriptor chain. If it is the end, the engine enters an inactive state and waits to be re-activated by software. If it did not reach the end of the chain, it progresses to the next descriptor, and so on.

30.3.2.2 RAID6 Recovery Mode

In RAID6 Recovery mode, the XOR engine is able to reconstruct the original data when either 1 or 2 of the data buffers P / Q are known to be invalid.The XOR engine reads the valid data buffers and writes the reconstructed data to the destination buffers. The parameters of the RAID6 Recovery operation are configured by writing the relevant information to a chain descriptor. The relevant parameters consist of buffers addresses, the number of total data buffers, and an indication of whether each buffer contains valid or invalid data, as well as various control information.

To use this mode, the user must write the following information to the chain descriptor:

- Specify the number of total buffers.
- For each buffer, specify the buffer address and the buffer validity.
- For P buffers specify the buffer address and the buffer validity.
- For Q buffers specify the buffer address and buffer validity.

There can be either 1 or 2 buffers that are defined as non-valid.

Upon starting the operation the XOR engine will read the data from the valid buffers and write the reconstructed data into the non-valid buffer locations.

Copyright © 2015 Marvell

The XOR engine supports the following reconstruction options:

- Reconstruct a single data buffer from other data buffers and P buffer.
- Reconstruct 2 data buffers from other data buffers and P and Q buffers.
- Reconstruct a single P buffer from data buffers.
- Reconstruct a single Q buffer from data buffers.
- Reconstruct P and Q buffers from data buffers.
- Reconstruct a data buffer and P buffer from other data buffers and Q buffer.
- Reconstruct a data buffer and Q buffer from other data buffers and P buffer.

30.3.2.3 RAID6 Multiply Mode

In RAID6 Multiply mode the XOR engine multiplies Source Buffer#0 by the G_Constant descriptor field, and adds to it the Source Buffers 1–7. The result is transferred to the Q buffer.

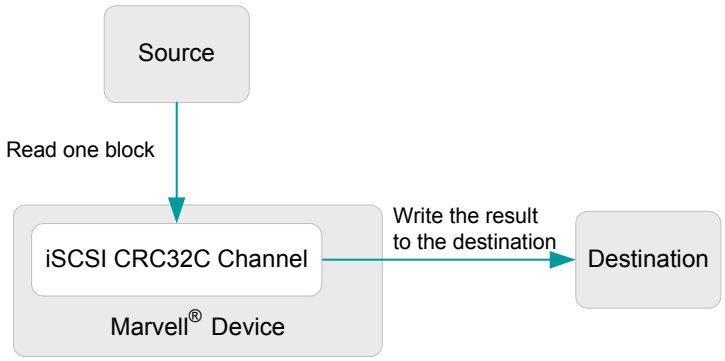
The calculations are performed over the GF(2^8) field.

This mode of operation may help implementing RAID6 for more than 8 disks.

30.3.3 iSCSI CRC32C Calculation

In addition to the XOR operation, the XOR engine also provides iSCSI CRC32C calculation capabilities. It performs iSCSI CRC32C calculation on a source block and writes the result back to a descriptor, as shown in Figure 90.

Figure 90: XOR iSCSI CRC32C Operation



The source blocks are specified through a chain of descriptors. Parameters of the iSCSI CRC32C operation are configured in the same way as in XOR operation - writing the relevant information to a chain descriptor. The relevant parameters consist of source addresses, destination address, size of source block, and ‘last block in CRC source chain’ indication.

The CRC source block in CRC mode can be scattered over a few target blocks. It can be in non-consecutive memory spaces and even in different interfaces. The CRC Source block is represented by a source block chain of descriptors. Every descriptor represents one consecutive section of the CRC source block.

When activated in CRC mode (see the <OperationMode> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1372) is set to CRC), XOR engine executes iSCSI CRC32C calculation according to the source block chain. The last descriptor in a source block chain is marked as ‘last’. After the calculation is finished, the 32-bit result is written to the CRC source block chain’s last descriptor.

The chain descriptor operation is slightly different in CRC-32 mode than in XOR mode. In XOR mode, every descriptor stands for a self contained XOR operation. In CRC mode, one CRC operation (one CRC source block) can be represented by a number of chain descriptors, thus

Copyright © 2015 Marvell

enabling concatenation of a few data sources to one block, for CRC calculation. The descriptor chain is constructed of several source block chains.

In all buffers used for iSCSI CRC calculation, align the data size to 4B.

The Basic iSCSI CRC32C operation is as follows:

1. Read data from the source block to internal buffer.
2. Calculate iSCSI CRC32C on the internal buffer and store intermediate result.
3. Repeat step 1-2 for the rest of the source block, until all of the blocks are processed.
4. Repeat steps 1-3 for the rest of the blocks in the source block chain.
5. Write result to the last descriptor.

30.3.4 DMA Operation

The XOR engine also provides generic DMA capabilities—copying of a source block to a destination block. The source blocks are specified through a chain of descriptors. Parameters of the DMA operation are configured in the same way as in XOR operation - by writing the relevant information to a chain descriptor. The relevant parameters consist of source address, destination address, and size of source block.

When activated in DMA mode, the XOR engine fetches the first descriptor and starts performing the DMA operation according to its parameters. After finishing the operation, the XOR engine closes the descriptor by writing back status word to the descriptor and returning the ownership of it to the CPU. The XOR engine checks whether it reached the end of the descriptor chain. If the end has been reached, the engine enters an inactive state and waits to be re-activated by the software. If it did not reach the end of the chain, it progresses to the next descriptor, and so on.

30.3.5 Memory Initialization

The XOR engine provides memory value initialization capabilities. It performs writes of pre-defined values to a destination memory block. The destination address and block size are specified directly by internal registers. The relevant parameters consist of a destination address, an initial memory value, and a size of the destination block.



Note

Only one channel may be configured to MemInit mode at a time. If both channels are configured to MemInit mode, engine behavior is unpredictable.

When activated in MemInit mode, the XOR engine executes a memory initialization operation according to the relevant internal registers. It will write the 64-bit initial value, specified by the XOR Engine Initial Value Low (XEIVRL) Register (m=0–1) (Table 1856 p. 1377) and XOR Engine Initial Value High (XEIVRH) Register (m=0–1) (Table 1857 p. 1377), in a cyclical method to the destination block. Upon completion of the memory initialization operation, the XOR engine channel asserts the EOC interrupt.

30.3.6 Memory ECC Errors Cleanup (Scrubbing)

The XOR engine provides single bit ECC error cleanup capabilities. It scans for ECC errors on a pre-defined destination block in DRAM. If a single bit error is detected, it is fixed. If two or more errors are detected (non-correctable ECC errors), an interrupt is asserted by the memory controller.

The memory ECC error cleanup operation is based on the device DRAM Read-Modify-Write (RMW) feature. Upon a write request to memory, with all byte enables inactive, the DRAM controller initiates an automatic RMW operation, without changing the data. If, during the read portion of the RMW, the controller detects a single bit error (or no error at all), it corrects the data and writes it back to memory. In case of a double-bit ECC error detection, the DRAM controller asserts an interrupt.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Parameters of the memory ECC error cleanup operation are configured in the same way as in memory initialization operation—by writing the relevant information to internal registers. The relevant parameters consist of the destination address and the destination block size.

Note

Only one channel may be configured to ECC mode at a time. If both channels are configured to ECC mode, engine behavior is unexpected.

In addition, the ECC operation supports a timer mode that enables periodic activation of the ECC cleanup operation, to a small target block at a time. This avoids long periods of memory usage by the ECC operation that can interfere with the normal system operation. Moreover, ECC error cleanup can run in the background, without CPU involvement. Once the CPU configures the parameters for the ECC operation, the ECC error cleanup takes place periodically, without any CPU intervention. The destination block is divided into sections according to the <SectionSizeCtrl> field in the XOR Engine Timer Mode Control (XETMCR) Register (m=0–1) (Table 1853 p. 1375).

The period between cleanup of sequential sectors is controlled by the ECC timer—a 32-bit wide timer integrated in the XOR engine. With every expiration of the ECC timer, another section of the target block is cleaned. Both channel0 and channel1 are coupled to the ECC timer. The timer decrements with every TCLK cycle. Upon expiration, cleanup of the relevant sector is initiated. The timer issues a timer expiration interrupt, reloads itself to the programmed initial value, and initiates a new count down. Reads from the timer are done from the counter itself, while writes are done to its register. This means that read results are in the counter's real-time value. The timer is enabled only if one of the XOR engine channels is set to ECC timer mode operation.

If activated in timer mode, the XOR engine:

- Enables timer count-down
- Waits for timer expiration
- Cleans the first memory section
- Asserts EOD interrupt

The XOR engine then waits for the second expiration, cleans the second section and asserts another EOD interrupt, and so on. When the XOR engine reaches the end of the target block, it asserts an EOC interrupt and re-starts cleaning the first memory section. To stop the operation, set the <XEstop> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1373).

If activated in non-timer mode, the whole target block is processed at one time and the XOR unit becomes inactive.

If the destination block is not 8 byte aligned, the XOR engine initiates write requests to the smallest 8 byte aligned block that contains the original block.

30.4 Descriptor Chain Format

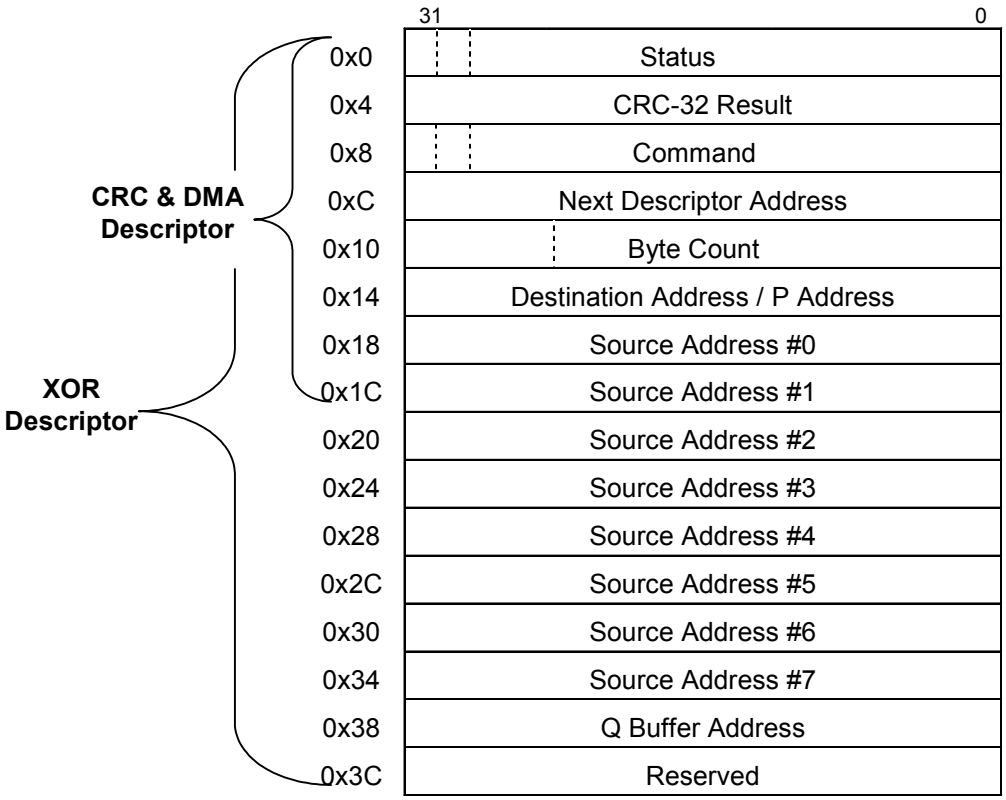
The XOR engine descriptor format supports 32-bit addressing. In XOR and RAID6 modes, the descriptor consists of sixteen 32-bit words which totals the 64B size of each descriptor. In CRC and DMA modes, only the upper 32B of the descriptor is needed. Therefore, the descriptor consists of eight 32-bit words, totaling to a 32B size for each descriptor (see Figure 91).

By fetching a descriptor from memory, the XOR engine gets all the information about the next operation to be performed. When the XOR engine finishes the operation associated with a descriptor, it closes the descriptor by updating the status word. This means the operation completed successfully and returns the ownership of the descriptor to the CPU.

Copyright © 2015 Marvell

Note The chain descriptor operation is valid only in XOR, RAID6, CRC, and DMA operation modes. In ECC and MemInit modes, the XOR engine receives the operation data directly from its internal registers.

Figure 91: XOR Descriptor Format



Data Alignment

- The XOR and RAID6 descriptors must be 64-byte aligned (Address[5:0]=0).
- The CRC and DMA descriptors must be 32-bytes aligned (Address[4:0]=0).
- There are no restrictions on source or destination data block alignment.
- Source and destination blocks can have different alignments.
- Different source blocks can have different alignments as well.

Document Classification: Proprietary Information

Table 82: Descriptor Status Word Definition

Bit	Field	Description
29:0	Reserved	Reserved.
30	Success	Successful descriptor execution indication. Indicates whether the operation completed successfully. 0 = Completed unsuccessfully. Transfer terminated before the whole byte count was transferred. 1 = Completed successfully. The whole byte count transferred. That field is updated upon closing the descriptor **NOTE:** In ECC cleanup mode the success bit indicates successful execution, even if ECC errors where found and not corrected.
31	Own	Ownership Bit Indicates whether the descriptor is owned by the CPU or the XOR engine. 0 = CPU owned. 1 = XOR engine owned. That field is updated upon closing a descriptor - XOR engine gives back ownership to the CPU by clearing the own bit.

Table 83: Descriptor CRC-32 Result Word Definition

Bit	Field	Description
31:0	CRCresult	Result of CRC-32 calculation Valid only in the last descriptor of a CRC source block chain, after it was closed by the XOR engine. **NOTE:** Valid only in CRC mode.

Table 84: Descriptor Command Word Definition

Bit	Field	Description
0	Src0Cmd	Specifies the type of operation to be carried out on the data pointed by SA#0 (Source Address 0 word of the descriptor). It is relevant only in XOR and RAID6 operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q Calculation, and RAID6 Multiply Operation modes: 0x0 = Data from the source is disregarded in the current descriptor operation. 0x1 = Data from the source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data, and it should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data.

Copyright © 2015 Marvell

Table 84: Descriptor Command Word Definition (Continued)

Bit	Field	Description
1	Src1Cmd	Specifies the type of operation to be carried out on the data pointed by SA#1 (Source Address 1 word of the descriptor). It is relevant only in XOR and RAID6* operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q calculation, and RAID6 multiply operation modes: 0x0 = Data from source is disregarded in the current descriptor operation. 0x1 = Data from source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 2.
2	Src2Cmd	Specifies the type of operation to be carried out on the data pointed by SA#2 (Source Address 2 word of the descriptor). It is relevant only in XOR and RAID6* operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q Calculation, and RAID6 Multiply Operation modes: 0x0 = Data from source is disregarded in the current descriptor operation. 0x1 = Data from source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 3.
3	Src3Cmd	Specifies the type of operation to be carried out on the data pointed by SA#3 (Source Address 3 word of the descriptor). It is relevant only in XOR and RAID6* operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q Calculation, and RAID6 Multiply Operation modes: 0x0 = Data from source is disregarded in the current descriptor operation. 0x1 = Data from source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 4.

Copyright © 2015 Marvell

Table 84: Descriptor Command Word Definition (Continued)

Bit	Field	Description
4	Src4Cmd	Specifies the type of operation to be carried out on the data pointed by SA#4 (Source Address 4word of the descriptor). It is relevant only on XOR and RAID6* operation modes. Disregarded in all other operation modes. For XOR, RAID6 P and QP and Q Calculation, RAID6 Multiply Operation modes: 0x0 = Data from Source is disregarded in the current descriptor operation. 0x1 = Data from source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 5.
5	Src5Cmd	Specifies the type of operation to be carried out on the data pointed by SA#5 (Source Address 5 word of the descriptor). It is relevant only on XOR and RAID6* operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q Calculation, and RAID6 Multiply Operation modes: 0x0 = Data from source is disregarded in the current descriptor operation. 0x1 = Data from source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 6.
6	Src6Cmd	Specifies the type of operation to be carried out on the data pointed by SA#6 (Source Address 6 word of the descriptor). It is relevant only in XOR and RAID6* operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q Calculation, and RAID6 Multiply Operation modes: 0x0 = Data from source is disregarded in the current descriptor operation. 0x1 = Data from source is transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and it should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 7.

Copyright © 2015 Marvell

Table 84: Descriptor Command Word Definition (Continued)

Bit	Field	Description
7	Src7Cmd	Specifies the type of operation to be carried out on the data pointed by SA#7 (Source Address 7 word of the descriptor). It is relevant only on XOR and RAID6* operation modes. It is disregarded in all other operation modes. For XOR, RAID6 P and Q Calculation, and RAID6 Multiply Operation modes: 0x0 = Data from source is disregarded in the current descriptor operation. 0x1 = Data from source will be transferred and is significant in the calculation. For RAID6 Recovery Operation mode: 0x0 = Data from this source is faulty data and it should be recovered. 0x1 = Data from this source is valid data and should be used for recovering faulty data of other buffers and/or generation of P/Q data. **NOTE:** In RAID6 Recovery Operation mode, this bit is valid only if the NoOfBuffers field >= 8.
8	PBufCmd	Specifies the type of operation to be carried out on the data pointed by P Buffer Address word of the descriptor. It is relevant only on RAID6 P and Q Generation and RAID6 Recovery Operation modes. It is disregarded in all other operation modes. In RAID6 P and Q Generation mode, this field controls generation of the P data: 0x0 = Do not generate P data. 0x1 = Generate P data. **NOTE:** In RAID6 P and Q Generation mode, at least 1 of the bits PBufCmd or QBufCmd should be set to 0x1. In RAID6 Recovery Operation mode: 0x0 = Data in this buffer does not contain valid P data. The P data should be recovered. 0x1 = Data in this buffer contains valid P data. It is used to recover data buffers and/or the Q buffer.
9	QBufCmd	Specifies the type of operation to be carried out on the data pointed by Q Buffer Address word of the descriptor. It is relevant only on RAID6 P and Q Generation and RAID6 Recovery Operation modes. It is disregarded in all other operation modes. In RAID6 P and Q Generation mode, this field controls generation of the Q data: 0x0 = Do not generate Q data. 0x1 = Generate Q data. **NOTE:** In RAID6 P and Q Generation mode, at least 1 of the bits PBufCmd or QBufCmd should be set to 0x1. In RAID6 Recovery Operation mode: 0x0 = Data in this buffer does not contain valid Q data. The Q data should be recovered. 0x1 = Data in this buffer contains valid Q data. It is used to recover data buffers and/or the P buffer.
11:10	Reserved	Reserved

Copyright © 2015 Marvell

Table 84: Descriptor Command Word Definition (Continued)

Bit	Field	Description
15:12	NumOfBuffers	This field is valid only in RAID6 Recovery Operation mode. It defines the number of data buffers. Valid values are 1–8. The pointers to the various data buffers are located in the Source Address#0 –7 fields.
23:16	GConstant	This field is valid only for RAID6 Multiply mode. It contains the data by which Source buffer#0 data is multiplied.
27:24	OperationMode	This field is valid only if the <OperationMode> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1372) is configured to 0x7 (Descriptor). Specifies the type of operation to be carried out by XOR Engine. 0x0 = XOR calculate operation 0x1 = CRC-32 calculate operation 0x2 = DMA operation 0x5 = RAID6 P and Q Generation: Generation of P and Q buffers data 0x6 = RAID6 Recovery: Performs recovery of the following (as indicated by Src*Cmd, PBufCmd, QBufCmd): 2 data buffers 1 data buffer + P buffer 1 data buffer + Q buffer 1 data buffer **NOTE:** If none of the above scenarios is identified, the P buffer is regenerated. 0x8 = RAID6 Multiply: Generation of G_Constant*Source_buffer#0 + Source_buffer#1 + … + Source_buffer#7. The result is stored in the Q buffer. Other values are reserved.
29:28	Reserved	Reserved
30	CRCLast	Indicated last descriptor in a CRC-32 calculation chain. 0 = Not last descriptor in a CRC calculation chain. 1 = Last descriptor in a CRC calculation chain. When closing the descriptor, the XOR engine writes the CRC result to its CRC-32 Result word. The next descriptor in the descriptor chain initiates a new CRC calculation. If the source block is represented by one descriptor only, it should be marked as last. **NOTE:** Relevant only in CRC operation mode.
31	EODIntEn	End Of Descriptor Interrupt Enable. Specifies if the EOD interrupt is asserted upon closure of that descriptor. 1 = EOD enabled. 0 = EOD disabled.

Table 85: Descriptor Next Descriptor Address Word

Bits	Field	Description
31:0	NDA	Next descriptor address pointer XOR and RAID6 modes: NDA must be 64-byte aligned (bits[5:0] must be 0x0). CRC/DMA Mode: NDA must be 32-byte aligned (bits[4:0] must be 0x0). NDA field of the last descriptor of a descriptor chain must be NULL.

Copyright © 2015 Marvell

Table 86: Descriptor Byte Count Word

Bit	Field	Description
23:0	ByteCount	XOR and RAID6 modes: Size of source and destination blocks in bytes. CRC mode: Size of source block part represented by the descriptor. DMA mode: Size of source and destination block in bytes. Minimum blocks' size: 1B. Maximum blocks' size: 16MB-1
31:24	Reserved	Reserved.

Table 87: Descriptor Destination Address Word

Bits	Field	Description
31:0	DA	Destination/P Block address pointer. XOR Mode: Destination Block address pointer. CRC mode: Not used. DMA/DMA_multiply mode: Destination Block address pointer. RAID6_P&Q_generation mode: P Block address pointer.

Table 88: Descriptor Source Address #N Words

Bits	Field	Description
31:0	SA#0 Source Address #0	Source block #0 address pointer. XOR Mode: Source Block #0 address pointer. CRC mode: Address pointer to part of source block represented by the descriptor. DMA Mode: Source Block address pointer. RAID6* modes: Address pointer to Buffer#0. For Q generation it is multiplied by g**0.
31:0	SA#N [N=1..7] Source Address #N	Source block #N address pointer. XOR Mode: Source Block #N address pointer. CRC mode: Not used. DMA mode: Not used. RAID6* modes: Address pointer to buffer#N. For Q generation it is multiplied by g**N.

Table 89: Descriptor Q Buffer Address Word

Bits	Field	Description
31:0	Q Buffer	Q Buffer address pointer. It is used only in RAID6 mode. It points to the buffer that contains the Q data.

Copyright © 2015 Marvell

30.5 Address Decoding

The XOR engine has eight address windows that can be individually configured. With each transaction, the XOR engine first compares the address (source, destination, or descriptor) against the address decoding registers. Each window can be configured to a different target interface. Address comparison is done to select the correct target interface. If the address does not match any of the address windows (no hit), an interrupt is generated and the XOR engine is stopped. If the address matches more than one address window (multiple hit), an interrupt is generated and the XOR engine is stopped.

For the XOR engine to avoid accessing forbidden address space (due to a programing bug), each channel uses access protection logic that prevents it from read/write access to specific address windows. In case of access violation, the operation is stopped, the channel becomes inactive, and an interrupt is asserted.

30.5.1 Target Interface

Source data blocks, destination data block, and descriptors can be targeted to any of the chip Interfaces. The unique attributes of each interface are configured per address window through the XOR engine BARs (Base Address Registers) (see Table 1838, Summary Map Table for the XOR Engine Registers, on page 1363).

30.5.2 64-bit Addressing

Four of the eight address windows have an upper 32-bit address register. These are used for accessing interfaces that support more than 4 GB of address space. The address generated on the interface is composed of the 32-bit address issued by the XOR engine, if it hits the relevant address window, concatenated with the High Remap register.

The XOR engine address decoder can map a total of up to a 4 GB address space.

30.5.3 Address Override

The XOR engine also supports an address override feature. Each of the sources, destination, or next descriptor addresses of each channel, can be configured to use the override feature by using the XOR Engine Address Override Control (XEAOCR)<n> Register (m=0–1, n=0–1) (Table 1843 p. 1368). When override is enabled and the respective pointer field is cleared to 0x0, the transaction target interface, and attributes, are taken from the Base Address register 0 (XEBAR0) and the upper 32 bits of the 64 bit address are taken from High Address Remap Register 0 (XEHARR0). When set to 0x1, these items are taken from XEBAR1and XEHARR1 respectively, and so on for pointer values of 0x2 and 0x3.

This address override feature, enables additional address de-coupling. For example, it allows the use of the same source and destination addresses, while the source is targeted to one interface and destination to a different interface.



Note

When using the address override option, no window access control is performed. For example if override is set to SA#5 of channel 1, any address that is specified in the descriptor as SA#5 is directly accessed without any window access control check.

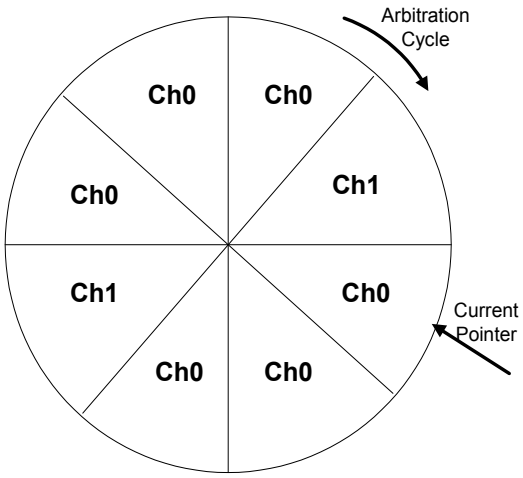
Copyright © 2015 Marvell

30.6 Arbitration

The device includes 4 XOR channels in 2 different units, 2 in each unit. The following section describes the arbitration scheme between each 2 channels.

A programmable weighted round robin arbiter controls the bandwidth allocation for each channel on the Mbus port. Each channel can be configured to have a different bandwidth allocation. Figure 92 shows an example of the arbitration cycle.

Figure 92: Programmable Channel Pizza Arbiter



The pizza arbiter has eight slices, each slice can be configured to serve a different channel. In Figure 92, channel0 gets 75% of the bandwidth, and channel1 25% each. At each clock cycle, the arbiter samples all channels requests and gives the bus to the next channel according to the “pizza” setting.

The bandwidth allocation is flexible. The arbiter influences the bandwidth allocation only when 2 ports demand Mbus port service at the same time (congestion conditions). If only 1 channel demands Mbus bandwidth, the channel receives 100% of the bandwidth. For example, in Figure 92, only Channel0 is active and so it gets 100% of the Mbus port bandwidth.

Copyright © 2015 Marvell

30.7 XOR Engine Programming

30.7.1 Programming in XOR, CRC, and DMA Modes

The XOR engine operation is similar in XOR, CRC, and DMA modes. All of these modes use descriptor chains. The modes differ in their configuration parameters and in their chain descriptor size and field interpretation.

30.7.1.1 Activation

To activate an XOR engine when it is inactive, the software must perform the following sequence:

1. Confirm that the relevant XOR engine channel is inactive (the <XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is cleared to 0).
2. Initialize the relevant XOR engine channel configuration through the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371).
3. Prepare the descriptor (or chain of descriptors) in memory.
4. Update the relevant XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1) (Table 1848 p. 1374) register.
5. Set the <XEStart> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1373).

When the XOR engine starts, the engine activates the relevant channel with the <XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is set.

When activated (<XEstatus> is set), the XOR engine fetches the descriptor pointed by the XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1) (Table 1848 p. 1374), and starts performing the operation on it. Upon completion of the operation it progresses to the next descriptor. It continues this operation until it reaches the end of the descriptor chain (Next descriptor Address field of current descriptor = NULL). When it reaches the end of the chain, the XOR engine asserts an interrupt (EOC - End Of Chain interrupt), clears the <XEstatus> bit and enters an inactive state. This inactive state is equal to the initial state of XOR engine upon startup.

30.7.1.2 Update Descriptor Chain

A new descriptor can be added to the chain when the XOR engine is active (<XEstatus>=1).

The software adds new descriptors to the descriptor chain by performing the following:

1. Prepares new descriptors (or chain of descriptors) in memory.
2. Updates the next descriptor address field in the former last descriptor.



Note

If the ownership mechanism is violated, the CPU will write to an XOR engine owned descriptor (the former last one). This does not affect the XOR engine operation.

30.7.1.3 Pause Operation

The pause operation enables a temporary halt of the current descriptor chain processing and then a continuation of it without any impact on the execution, except the delay caused by the pause period. When paused the XOR engine channel does not initiate any requests to the Mbus, releasing its resources to other units.

The pause operation can be used for boosting performance of a mission critical process for a specific time period. After the critical time period is over, the software signals the XOR engine channel to continue processing the current descriptor chain from the point at which it was paused.

Copyright © 2015 Marvell

The software can pause the XOR engine channel operation during an active phase by performing the following:

1. Confirm that the relevant XOR engine channel is active (<XEpause> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is set). If it is not active, the pause operation is not necessary.
2. Set the relevant <XEpause>.
3. Check the relevant <XEstatus> field. When it is cleared, the pause operation completed.

When paused (<XEpause> is set), the XOR engine channel suspends the current operation at the earliest opportunity, and enters a pause state. Upon entering a pause state, the XOR engine channel signals the software by clearing the <XEstatus> bit in the activation register and asserting the paused interrupt.

Receipt of an EOC interrupt before a paused interrupt, after initiation of a pause operation, implies that:

- The channel completed the current descriptor chain before the pause operation.
- The channel is in stop mode and not in paused mode.

The software must act accordingly and reactivate the channel according to the Re-Activation After Stop information.

Re-Activation After Pause

To re-activate the channel, the software must set the <XErestart> field in the relevant activation register (XE0ACTR or XE1ACTR). When <XEstatus> field is set, the XOR engine has resumed operation.

After pausing a channel, it is not allowed to stop it. To stop the channel, the software must first perform a re-activation after pause operation. Only after the channel becomes active can it be stopped.

30.7.1.4 Stop Operation

The stop operation terminates processing of an XOR engine channel's current operation. After stop, the current operation cannot be resumed and a new operation must be loaded to the XOR engine channel. The software can stop the XOR engine channel operation while active, by performing the following:

1. Check that the relevant XOR engine channel is active (<XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is set). If it is not active, the stop operation is not necessary.
2. Set the relevant <XEstop> field.
3. Check the relevant <XEstatus> field. When it is cleared, the stop operation is completed.

When stopped (<XEstop> is set), the XOR engine stops performing the operation at the earliest opportunity and enters an inactive state. Upon entering an inactive state, the XOR engine closes the current descriptor and signals the software by clearing the <XEstatus> field in the activation register and asserting the stopped interrupt. Inactive state is similar to Initial state of XOR engine on startup.

Re-Activation After Stop

Re-activation is similar to the activation procedure in Section 30.7.1.1, Activation, on page 345.

The software must perform the following steps:

1. Confirm that the relevant XOR engine channel is inactive. The <XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is cleared to 0.
2. Initialize the relevant XOR engine channel through the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371).

Document Classification: Proprietary Information
Copyright © 2015 Marvell

3. Prepare a descriptor (or chain of descriptors) in memory.
4. Update the XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1) (Table 1848 p. 1374).
5. Set the <XErestart> field.

When the XOR engine starts, the engine activates the relevant channel with the <XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is set.

When activated (<XEstatus> is set), the XOR engine fetches the descriptor pointed by the XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1) (Table 1848 p. 1374), and starts performing the operation on it. Upon completion, it progresses to the next descriptor and continues until the end of the descriptor chain is reached, the EOC - Next descriptor Address field of the current descriptor equals NULL. Upon reaching the end of the chain, the XOR engine clears the <XEstatus> bit and enters inactive state. This state is equal to the initial state of XOR engine on startup.

30.7.1.5 Reaching End of Descriptor Chain

Upon reaching the end of the descriptor chain, the XOR engine asserts an EOC (End Of Chain) interrupt and enters an inactive state. It waits to be re-activated by the software (setting <XEStart> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1373)).

Upon receiving an EOC interrupt, two options must be examined by the software:

True EOC XOR engine reaches the end of a descriptor chain.

False EOC Chain was updated and the XOR engine is not in the current EOC. This can occur when software updates the descriptor chain while the XOR engine is processing the former last descriptor in the chain. In this case, the XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1) (Table 1848 p. 1374) has a NULL value. Although it did not reach a true EOC, the XOR engine enters an inactive state.

To determine which option is valid, and to act accordingly, the software must check if the XOR engine current descriptor is currently the last descriptor in the chain. For example, read the XOR Engine Current Descriptor Pointer (XEnCDPR)<n> Register (m=0–1, n=0–1) (Table 1849 p. 1374) and match it with the software's current descriptor parameter.

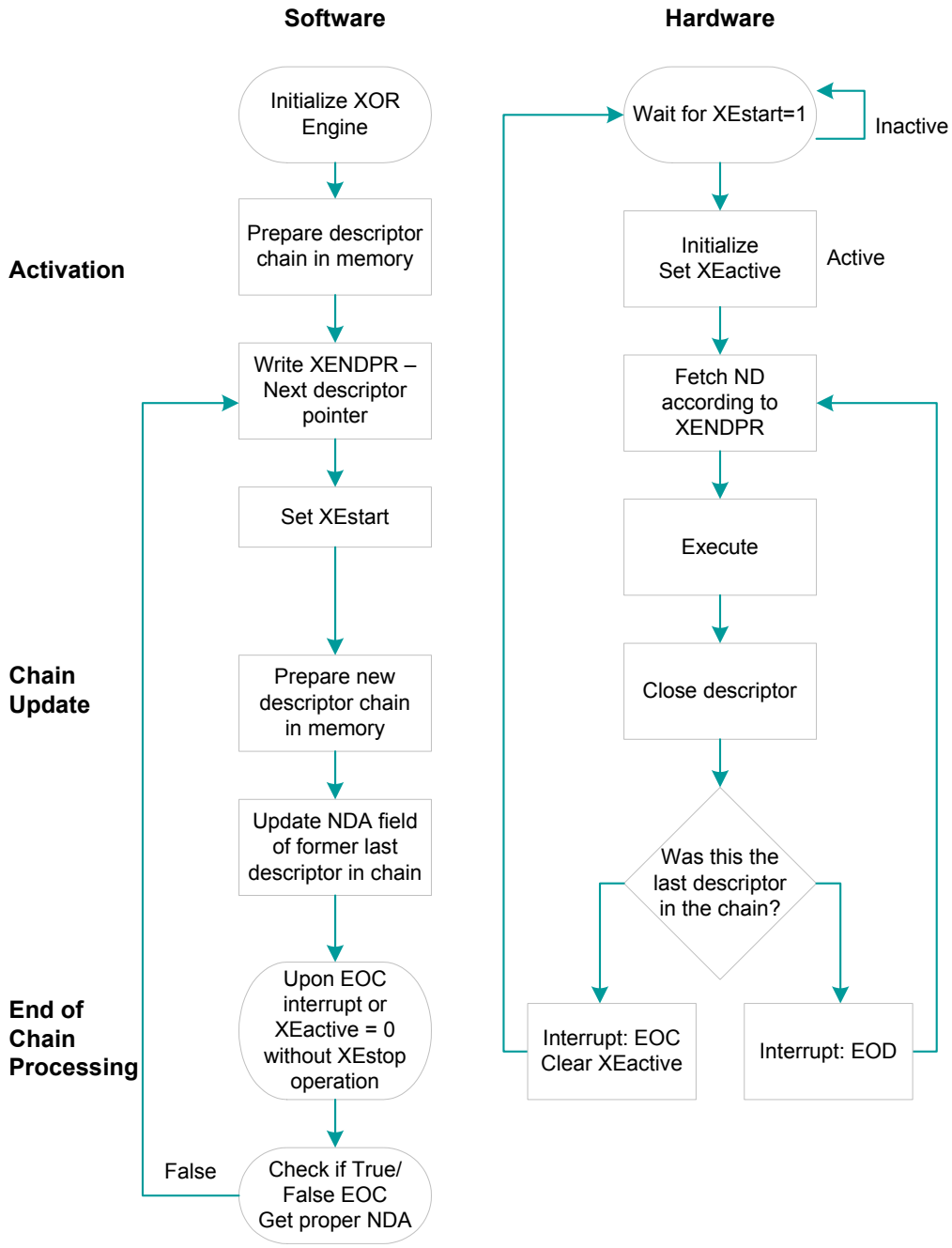
If it is true, EOC acts according to activation after stop.

If it is false EOC forces the XOR engine to re-read the current descriptor. That is done by writing to the current descriptor pointer, that was read from XECDPR, to the Next descriptor Pointer Register (XENDPR), and performing an activation after stop, set the <XEstart> bit in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372).

Copyright © 2015 Marvell

30.7.1.6 Synchronizing Software and Hardware

Figure 93: Software and Hardware Synchronization



Copyright © 2015 Marvell

30.7.2 Programming in ECC and MemInit Modes

The ECC and MemInit modes are programmed and controlled directly through internal registers (without using descriptor chains).

30.7.2.1 Activation

To activate the XOR engine, the software must perform the following sequence:

1. Confirm that XOR engine relevant channel is inactive. The <XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372) is cleared to 0.
2. Initialize the relevant XOR engine channel configuration through the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371).
3. Program the relevant internal registers (XOR engine ECC/MemInit Registers).
4. Set the <XErestart> field.

30.7.2.2 Stop Operation

The stop operation terminates a XOR engine channel's processing of the current operation. After stop, the current operation cannot be resumed. A new operation must be loaded to the XOR engine channel.

To stop the XOR engine channel operation while active, performing the following:

1. Check that the relevant XOR engine channel is active. The <XEstatus> field in the XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Table 1846 p. 1372)Register must be set. If it is not active, the stop operation is not necessary.
2. Set the <XEstop> field in the relevant activation register.
3. Check the relevant <XEstatus> field. When it is cleared, the stop operation is completed.

When stopped (<XEstop> is set), the XOR engine stops performing the operation at the earliest opportunity and enters an inactive state. Upon entering inactive state, the XOR engine signals the software by clearing the <XEstatus> field in the activation register and asserting the stopped interrupt. The inactive state is similar to initial state of the XOR engine on startup.

30.7.3 Internal Registers Write Access Protection

When an XOR engine channel is active, all the registers that are related to that channel, the shared address decoding registers, the shared channel arbitration registers and the shared memory initialization initial value registers, are write access protected. Every write request to those internal registers when the channel is active (<XEstatus> of the relevant channel is set) is silently disregarded. The only channel related registers that can be write accessed during the channel active period are the activation registers, the shared interrupt cause and mask registers, and the debug register.

This design prevents configuration changes during channel operation. Changes during a channel's operation can cause unpredictable results. The register access protection can be de-activated per channel through the relevant <RegAccProtect> field in the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371).

If at least one of the channels enables register write access protection, write accesses to internal registers shared by channels (for example, address decoding, channel arbitration, and memory initialization initial value registers) are disregarded.

Read requests for all internal registers are enabled at all times, regardless of the channel activation status (except for WO - Write Only registers).

Copyright © 2015 Marvell

30.8 Burst Limit

The maximum burst sizes of different transaction types on the Mbus can be configured through the XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Table 1845 p. 1371).

- Data read (reading a source buffer) maximum burst size: 32B/64B/128B.
- Data write (writing to destination buffer) maximum burst size: 32B/64B/128B.

A descriptor read, fetching a descriptor, is always a 32B burst size. In XOR mode, almost all the 64 bytes of the descriptor are relevant and two 32B read requests are required. In CRC or ECC modes, only the upper 32B of the descriptor are relevant and one 32B read request is sufficient.

Descriptor write, closing a descriptor, is always 8B burst size (Status and iSCSI CRC32C Result words).

The XOR Engine also supports limiting the number of data fetch outstanding requests. This mode can be used to reduce the throughput of the XOR engine, in case it interferes with other units in the device.

To configure the number of outstanding reads, set the appropriate configuration in the <MaxNoOfOutstandingReads> field in the XOR Engine Outstanding Reads (XEnORDS)<n> Register (m=0–1, n=0–1) (Table 1847 p. 1373).

30.9 Errors and Interrupts

The XOR engine interrupts are registered in the XOR Engine Interrupt Cause (XEICR1) Register (m=0–1) (Table 1858 p. 1377). Upon an interrupt event, the corresponding cause bit is set to 1. It is cleared upon a software write of 0.

The XOR Engine Interrupt Mask (XEIMR) Register (m=0–1) (Table 1859 p. 1380) controls whether an interrupt event causes an interrupt assertion. The setting of the mask register only affects the interrupt assertion. This setting has no effect on the cause register bits setting.

The XOR engine interrupts are divided into 2 groups:

Error Interrupts — Descriptor ownership violation, address miss, multiple hit, window access violation, write protect violation, or parity error.

Operation Completion Interrupts — EOD (End of Descriptor), EOC (End of Chain), pause, or stop by software

Table 90 summarizes the interpretation of EOD and EOC interrupts for each operation mode.

Table 90: EOC/EOD interpretation

Operation Mode	Operation Related Interrupt Description
XOR, CRC, DMA, RAID6	• The EOD interrupt is asserted upon closing each descriptor. If the <EODIntEn> bit of the descriptor is cleared, an EOD interrupt is not asserted when it is closed. • The EOC interrupt is asserted upon reaching end of descriptor chain or upon end of chain processing due to error condition.
MemInit	• The EOC interrupt is asserted upon completing the MemInit operation.
ECC - Non-Timer mode	• The EOC interrupt is asserted after the entire destination block is cleaned.
ECC - Timer mode	• The EOD interrupt is asserted after every section cleanup completion. • The EOC interrupt is asserted after the entire destination block is cleaned.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

The following error interrupts are supported:

- Parity error: Internal data path parity error.
- Ownership error: Fetching descriptor that is owned by the CPU (software error).
- Address Miss Error: Accessing an address that is not in one of the address windows, or an address which matches more than one address window.
- Access Protect Error: Accessing an address which is access protected.
- Write Protect Error: Writing to a write protected address.

In all error conditions, the XOR engine halts, as if it is stopped by the software. Also, in all case of an error address, the address is latched in the XOR Engine Error Address (XEEAR) Register (m=0–1) (Table 1861 p. 1381). Once an address is latched, no new addresses (due to additional errors) can be latched until the current address being read.

30.10 Performance Optimization

To increase XOR engine performance:

- Set the source and destination buffers alignment according to the relevant maximum burst size configuration parameters (<SrcBurstLimit> and <DstBurstLimit>). For example, if SrcBurstLimit of Channel0 is 128B and DstBurstLimit is 32B, it is recommended to keep the source buffers in 128B aligned addresses and the destination buffers in 32B aligned addresses.
- Set the buffer sizes as large as possible.

Copyright © 2015 Marvell

31 Electronic Fuse (eFuse)

This section describes the Electronic Fuse (eFuse) functionality and interface. The eFuse is a one-time-programming (OTP) module.

31.1 Features

The device integrates:

- 2 low-density (LD) eFuse modules each containing 256b of data for a total of 512b.
  - 1x LD eFuse (LD0) is reserved for Marvell internal use.
  - 1x LD eFuse (LD1) is available for storing general purpose information.
  - The LD fuse content is readable following power up.
- 1x high-density (HD) eFuse module. The HD eFuse content is readable following power up.
- Each eFuse line contains a locking bit for data protection from future programming attempts.
- Each eFuse line has Soft Error protection.

Copyright © 2015 Marvell



88F6810, 88F6811, 88F6821, 88F6W21, 88F6820, and 88F6828

ARMADA® 38x Family
High-Performance Single/Dual CPU System on Chip

Functional Specifications – Unrestricted

Appendix A: Register Set

Doc. No. MV-S109094-U0, Rev. A
November 25, 2015, Preliminary

Document Classification: Proprietary Information

THIS PAGE IS INTENTIONALLY LEFT BLANK

Copyright © 2015 Marvell

List of Registers

A.2 CPU Subsystem Registers 439

Table 93: IO Window 0 Control Register 445
Offset: 0x00020000

Table 94: IO Window 0 Base Register 445
Offset: 0x00020004

Table 95: IO Window 0 Remap Low Register 446
Offset: 0x00020008

Table 96: IO Window 0 Remap High Register 446
Offset: 0x0002000C

Table 97: IO Window 1 Control Register 446
Offset: 0x00020010

Table 98: IO Window 1 Base Register 447
Offset: 0x00020014

Table 99: IO Window 1 Remap Low Register 447
Offset: 0x00020018

Table 100: IO Window 1 Remap High Register 447
Offset: 0x0002001C

Table 101: IO Window 2 Control Register 447
Offset: 0x00020020

Table 102: IO Window 2 Base Register 448
Offset: 0x00020024

Table 103: IO Window 2 Remap Low Register 448
Offset: 0x00020028

Table 104: IO Window 2 Remap High Register 448
Offset: 0x0002002C

Table 105: IO Window 3 Control Register 448
Offset: 0x00020030

Table 106: IO Window 3 Base Register 449
Offset: 0x00020034

Table 107: IO Window 3 Remap Low Register 449
Offset: 0x00020038

Table 108: IO Window 3 Remap High Register 449
Offset: 0x0002003C

Table 109: IO Window 4 Control Register 449
Offset: 0x00020040

Table 110: IO Window 4 Base Register 450
Offset: 0x00020044

Table 111: IO Window 4 Remap Low Register 450
Offset: 0x00020048

Table 112: IO Window4 Remap High Register 450
Offset: 0x0002004C

Table 113: IO Window 5 Control Register 450
Offset: 0x00020050

Copyright © 2015 Marvell

Table 114: IO Window 5 Base Register 451
Offset: 0x00020054
Table 115: IO Window 5 Remap Low Register 451
Offset: 0x00020058
Table 116: IO Window 5 Remap High Register 451
Offset: 0x0002005C
Table 117: IO Window 6 Control Register 451
Offset: 0x00020060
Table 118: IO Window 6 Base Register 452
Offset: 0x00020064
Table 119: IO Window 6 Remap Low Register 452
Offset: 0x00020068
Table 120: IO Window 6 Remap High Register 452
Offset: 0x0002006C
Table 121: IO Window 7 Control Register 452
Offset: 0x00020070
Table 122: IO Window 7 Base Register 453
Offset: 0x00020074
Table 123: IO Window 7 Remap Low Register 453
Offset: 0x00020078
Table 124: IO Window 7 Remap High Register 453
Offset: 0x0002007C
Table 125: Internal Registers Base Address 453
Offset: 0x00020080
Table 126: Internal Units Sync Barrier Control Register 453
Offset: 0x00020084
Table 127: IO Window 8 Control Register 454
Offset: 0x00020090
Table 128: IO Window 8 Base Register 454
Offset: 0x00020094
Table 129: IO Window 9 Control Register 454
Offset: 0x00020098
Table 130: IO Window 9 Base Register 455
Offset: 0x0002009C
Table 131: IO Window 10 Control Register 455
Offset: 0x000200A0
Table 132: IO Window 10 Base Register 456
Offset: 0x000200A4
Table 133: IO Window 11 Control Register 456
Offset: 0x000200A8
Table 134: IO Window 11 Base Register 456
Offset: 0x000200AC
Table 135: IO Window 12 Control Register 456
Offset: 0x000200B0
Table 136: IO Window 12 Base Register 457
Offset: 0x000200B4
Table 137: IO Window 13 Control Register 457
Offset: 0x000200B8

Copyright © 2015 Marvell

Table 138: IO Window 13 Base Register 458
Offset: 0x000200BC
Table 139: IO Window 14 Control Register 458
Offset: 0x000200C0
Table 140: IO Window 14 Base Register 458
Offset: 0x000200C4
Table 141: IO Window 15 Control Register 458
Offset: 0x000200C8
Table 142: IO Window 15 Base Register 459
Offset: 0x000200CC
Table 143: IO Window 16 Control Register 459
Offset: 0x000200D0
Table 144: IO Window 16 Base Register 459
Offset: 0x000200D4
Table 145: IO Window 17 Control Register 460
Offset: 0x000200D8
Table 146: IO Window 17 Base Register 460
Offset: 0x000200DC
Table 147: IO Window 18 Control Register 460
Offset: 0x000200E0
Table 148: IO Window 18 Base Register 461
Offset: 0x000200E4
Table 149: IO Window 19 Control Register 461
Offset: 0x000200E8
Table 150: IO Window 19 Base Register 461
Offset: 0x000200EC
Table 151: IO Window 13 Remap Low Register 462
Offset: 0x000200F0
Table 152: IO Window 13 Remap High Register 462
Offset: 0x000200F4
Table 153: DRAM Window 0 Base Address Register 462
Offset: 0x00020180
Table 154: DRAM Window 0 Control Register 462
Offset: 0x00020184
Table 155: DRAM Window 1 Base Address Register 463
Offset: 0x00020188
Table 156: DRAM Window 1 Control Register 463
Offset: 0x0002018C
Table 157: DRAM Window 2 Base Address Register 464
Offset: 0x00020190
Table 158: DRAM Window 2 Control Register 464
Offset: 0x00020194
Table 159: DRAM Window 3 Base Address Register 465
Offset: 0x00020198
Table 160: DRAM Window 3 Control Register 465
Offset: 0x0002019C
Table 161: SoC Window Control Register 465
Offset: 0x00020250

Copyright © 2015 Marvell

Table 162: SoC Window Base Register 466
Offset: 0x00020254
Table 163: IO Error Control Register 466
Offset: 0x00020200
Table 164: PD Adapter Configuration Register 466
Offset: 0x00020220
Table 165: CPU Subsystem Error Status Register 467
Offset: 0x00020258
Table 166: CPU Subsystem Error Mask Register 469
Offset: 0x0002025C
Table 167: CPU Subsystem Local Cause Register 469
Offset: 0x00020260
Table 168: CIB Control and Configuration Register 469
Offset: 0x00020280
Table 169: CIB Non L1 Snoopable Window 0 Control Register 471
Offset: 0x00020284
Table 170: CIB Non L1 Snoopable Window 0 Base Address Register 471
Offset: 0x00020288
Table 171: CIB Non L1 Snoopable Window 1 Control Register 471
Offset: 0x0002028C
Table 172: CIB Non L1 Snoopable Window 1 Base Address Register 471
Offset: 0x00020290
Table 173: CIB Read Buffer Select Register 472
Offset: 0x00020294
Table 174: CIB Parity Error Status Register 472
Offset: 0x00020298
Table 175: SoC Global Timers Control Register 472
Offset: 0x00020300
Table 176: SoC Global Timers Events Status register Register 475
Offset: 0x00020304
Table 177: SoC Global Timer 0 Reload Register 475
Offset: 0x00020310
Table 178: SoC Global Timer 0 Register 476
Offset: 0x00020314
Table 179: SoC Global Timer 1 Reload Register 476
Offset: 0x00020318
Table 180: SoC Global Timer 1 Register 476
Offset: 0x0002031C
Table 181: SoC Global Timer 2 Reload Register 476
Offset: 0x00020320
Table 182: SoC Global Timer 2 Register 477
Offset: 0x00020324
Table 183: SoC Global Timer 3 Reload Register 477
Offset: 0x00020328
Table 184: SoC Global Timer 3 Register 477
Offset: 0x0002032C
Table 185: Shared Inbound Doorbell0 Register 477
Offset: 0x00020400

Copyright © 2015 Marvell

Table 186: Shared Inbound Doorbell1 Register 477
Offset: 0x00020404
Table 187: Shared Inbound Doorbell2 Register 478
Offset: 0x00020408
Table 188: Mbus Units Priority Control Register 478
Offset: 0x00020420
Table 189: Fabric Units Priority Control Register 480
Offset: 0x00020424
Table 190: Mbus Units Prefetch Control Register 480
Offset: 0x00020428
Table 191: Memory Management Unit (MMU) Static WindowN Register (N=0–7) 482
Offset: Windows0: 0x00020890, Windows1: 0x00020894, Windows2: 0x00020898, Windows3: 0x0002089C, Windows4: 0x000208A0, Windows5: 0x000208A4, Windows6: 0x000208A8, Windows7: 0x000208AC
Table 192: Semaphore0 Register 482
Offset: 0x00020500
Table 193: Semaphoren Register (n=1–7) 483
Offset: Semaphore1: 0x00020504, Semaphore2: 0x00020508, Semaphore3: 0x0002050C, Semaphore4: 0x00020510, Semaphore5: 0x00020514, Semaphore6: 0x00020518, Semaphore7: 0x0002051C
Table 194: Global Scratch Pad i Register (i=0–7) 484
Offset: pad id0: 0x00020520, pad id1: 0x00020540, pad id2: 0x00020560, pad id3: 0x00020580, pad id4: 0x000205A0, pad id5: 0x000205C0, pad id6: 0x000205E0, pad id7: 0x00020600
Table 195: Watchdog (WD) RSTOUTn Mask Register 484
Offset: 0x00020704
Table 196: Watchdog (WD) Interrupt Mask Register 484
Offset: 0x00020708
Table 197: CPUn Software Reset Control Register (n=0–1) 485
Offset: CPU ID0: 0x00020800, CPU ID1: 0x00020808
Table 198: Endpoint Interrupt Cause Vec0 Register 485
Offset: 0x00020900
Table 199: Endpoint Interrupt Cause Vec1 Register 485
Offset: 0x00020904
Table 200: Endpoint Interrupt Cause Vec2 Register 486
Offset: 0x00020908
Table 201: Endpoint Interrupt Cause Vec3 Register 486
Offset: 0x0002090C
Table 202: Endpoint Select Cause Register 486
Offset: 0x00020928
Table 203: Endpoint Fabric Local Interrupt Mask Register 487
Offset: 0x00020940
Table 204: Endpoint SoC Main Interrupt Error 0 Mask Register 487
Offset: 0x00020944
Table 205: Endpoint SoC Main Interrupt Error 1 Mask Register 487
Offset: 0x00020948
Table 206: Software Triggered Interrupt Register 488
Offset: 0x00020A04
Table 207: SoC Main Interrupt Error 0 Cause Register 489
Offset: 0x00020A20

Table 208: SoC Main Interrupt Error 1 Cause Register 489
Offset: 0x00020A24
Table 209: Endpoint ISE Register 489
Offset: 0x00020A30
Table 210: Endpoint ICE Register 489
Offset: 0x00020A34
Table 211: Interrupt Source i Control Register (i=0–28) 490
Offset: Interrupt ID0: 0x00020B00, Interrupt ID1: 0x00020B04...Interrupt ID28: 0x00020B70
Table 212: Endpoint Source i Control Register (i=29–115) 491
Offset: Interrupt ID29: 0x00020B74, Interrupt ID30: 0x00020B78...Interrupt ID115: 0x00020CCC
Table 213: CoreSight Components Base Address Register 492
Offset: 0x00020D00
Table 214: CoreSight Status Register 492
Offset: 0x00020D0C
Table 215: CA9 MP Core Reset Control Register 492
Offset: 0x00020D84
Table 216: CPU Configuration Register (N=0–1) 493
Offset: CPU ID0: 0x00021800, CPU ID1: 0x00021900
Table 217: CPU Control and Status Register (N=0–1) 494
Offset: CPU ID0: 0x00021808, CPU ID1: 0x00021908
Table 218: CPU IO Sync Barrier Control Register (N=0–1) 495
Offset: CPU ID0: 0x00021810, CPU ID1: 0x00021910
Table 219: SoC Private Timers Control Register (N=0–1) 495
Offset: CPU ID0: 0x00021840, CPU ID1: 0x00021940
Table 220: SoC Private Timer 0 Reload Register (N=0–1) 496
Offset: CPU ID0: 0x00021850, CPU ID1: 0x00021950
Table 221: SoC Private Timer 0 Register (N=0–1) 497
Offset: CPU ID0: 0x00021854, CPU ID1: 0x00021954
Table 222: SoC Private Timer 1 Reload Register (N=0–1) 497
Offset: CPU ID0: 0x00021858, CPU ID1: 0x00021958
Table 223: SoC Private Timer 1 Register (N=0–1) 497
Offset: CPU ID0: 0x0002185C, CPU ID1: 0x0002195C
Table 224: SoC Private Timers Events Status Register (N=0–1) 497
Offset: CPU ID0: 0x00021868, CPU ID1: 0x00021968
Table 225: Outbound Doorbell Register (N=0–1) 498
Offset: CPU ID0: 0x00021870, CPU ID1: 0x00021970
Table 226: Outbound Doorbell Mask Register (N=0–1) 498
Offset: CPU ID0: 0x00021874, CPU ID1: 0x00021974
Table 227: Inbound Doorbell Register (N=0–1) 499
Offset: CPU ID0: 0x00021878, CPU ID1: 0x00021978
Table 228: Inbound Doorbell Mask Register (N=0–1) 499
Offset: CPU ID0: 0x0002187C, CPU ID1: 0x0002197C
Table 229: SoC PPI Cause Reg Register (N=0–1) 499
Offset: CPU ID0: 0x00021880, CPU ID1: 0x00021980
Table 230: SoC Private Peripheral Interrupt Set Mask (PPI ISM) Register (N=0–1) 500
Offset: CPU ID0: 0x000218B8, CPU ID1: 0x000219B8
Table 231: SoC Private Peripheral Interrupt Clear Mask (PPI ICM) Register (N=0–1) 500
Offset: CPU ID0: 0x000218BC, CPU ID1: 0x000219BC

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 232: SoC Main Interrupt Error 0 Mask Register (N=0–1) 500
Offset: CPU ID0: 0x000218C0, CPU ID1: 0x000219C0
Table 233: SoC Main Interrupt Error 1 Mask Register (N=0–1) 501
Offset: CPU ID0: 0x000218C8, CPU ID1: 0x000219C8
Table 234: Local Scratch Pad 0 Register (N=0–1) 501
Offset: CPU ID0: 0x000218D0, CPU ID1: 0x000219D0
Table 235: Local Scratch Pad 1 Register (N=0–1) 501
Offset: CPU ID0: 0x000218D4, CPU ID1: 0x000219D4
Table 236: Local Scratch Pad 2 Register (N=0–1) 501
Offset: CPU ID0: 0x000218D8, CPU ID1: 0x000219D8
Table 237: Local Scratch Pad 3 Register (N=0–1) 501
Offset: CPU ID0: 0x000218DC, CPU ID1: 0x000219DC
Table 238: Local Scratch Pad 4 Register (N=0–1) 502
Offset: CPU ID0: 0x000218E0, CPU ID1: 0x000219E0
Table 239: Local Scratch Pad 5 Register (N=0–1) 502
Offset: CPU ID0: 0x000218E4, CPU ID1: 0x000219E4
Table 240: Local Scratch Pad 6 Register (N=0–1) 502
Offset: CPU ID0: 0x000218E8, CPU ID1: 0x000219E8
Table 241: Local Scratch Pad 7 Register (N=0–1) 502
Offset: CPU ID0: 0x000218EC, CPU ID1: 0x000219EC
Table 242: CPU Status Register (N=0–1) 502
Offset: CPU ID0: 0x000218F0, CPU ID1: 0x000219F0
Table 243: CPU External Debug Register (N=0–1) 503
Offset: CPU ID0: 0x000218F4, CPU ID1: 0x000219F4
Table 244: CPU Data Engine Flags Register (N=0–1) 503
Offset: CPU ID0: 0x000218F8, CPU ID1: 0x000219F8
Table 245: CPU Event Control Register (N=0–1) 503
Offset: CPU ID0: 0x000218FC, CPU ID1: 0x000219FC
Table 246: Timestamp Counter Enable Register 504
Offset: 0x00023000
Table 247: Timestamp Counter Preload 1 Register 504
Offset: 0x00023004
Table 248: Timestamp Counter Preload 2 Register 504
Offset: 0x00023008
Table 249: Timestamp Claim Tag Set Register 504
Offset: 0x00023FA0
Table 250: Timestamp Claim Tag Clear Register 504
Offset: 0x00023FA4
Table 251: Timestamp Lock Access Register 505
Offset: 0x00023FB0
Table 252: Timestamp Lock Status Register 505
Offset: 0x00023FB4
Table 253: Timestamp Authentication Status Register 505
Offset: 0x00023FB8
Table 254: Timestamp Device ID Register 505
Offset: 0x00023FC8
Table 255: Timestamp Device Type Identifier Register 505
Offset: 0x00023FCC

Copyright © 2015 Marvell

Table 256: Timestamp Peripheral ID4 Register 505
Offset: 0x00023FD0
Table 257: Timestamp Peripheral ID5 Register 506
Offset: 0x00023FD4
Table 258: Timestamp Peripheral ID6 Register 506
Offset: 0x00023FD8
Table 259: Timestamp Peripheral ID7 Register 506
Offset: 0x00023FDC
Table 260: Timestamp Peripheral ID0 Register 506
Offset: 0x00023FE0
Table 261: Timestamp Peripheral ID1 Register 506
Offset: 0x00023FE4
Table 262: Timestamp Peripheral ID2 Register 506
Offset: 0x00023FE8
Table 263: Timestamp Peripheral ID3 Register 507
Offset: 0x00023FEC
Table 264: Timestamp Component ID0 Register 507
Offset: 0x00023FF0
Table 265: Timestamp Component ID1 Register 507
Offset: 0x00023FF4
Table 266: Timestamp Component ID2 Register 507
Offset: 0x00023FF8
Table 267: Timestamp Component ID3 Register 507
Offset: 0x00023FFC

A.3 CPU Power Management Unit Registers 508

Table 269: CPU0 Statistic Counter Global Control Register 508
Offset: 0x00022100
Table 270: CPU0 PM Control and Configuration Register 509
Offset: 0x00022104
Table 271: CPU0 PM Status and Mask Register 509
Offset: 0x0002210C
Table 272: CPU0 Idle Timer Value Register 510
Offset: 0x00022110
Table 273: CPU0 Idle High Threshold Register 510
Offset: 0x00022114
Table 274: CPU0 Idle Low Threshold Register 511
Offset: 0x00022118
Table 275: CPU0 Idle Result Register 511
Offset: 0x0002211C
Table 276: CPU0 PM Event Status and Mask Register 511
Offset: 0x00022120
Table 277: CPU0 Boot Address Redirect Register 512
Offset: 0x00022124
Table 278: CPU0 Resume Control Register 512
Offset: 0x00022128
Table 279: CPU1 Statistic Counter Global Control Register 512
Offset: 0x00022200

Copyright © 2015 Marvell

Table 280: CPU1 PM Control and Configuration Register 513
Offset: 0x00022204
Table 281: CPU1 PM Status and Mask Register 513
Offset: 0x0002220C
Table 282: CPU1 Idle Timer Value Register 514
Offset: 0x00022210
Table 283: CPU1 Idle High Threshold Register 514
Offset: 0x00022214
Table 284: CPU1 Idle Low Threshold Register 515
Offset: 0x00022218
Table 285: CPU1 Idle Result Register 515
Offset: 0x0002221C
Table 286: CPU1 PM Event Status and Mask Register 515
Offset: 0x00022220
Table 287: CPU1 Boot Address Redirect Register 516
Offset: 0x00022224
Table 288: CPU1 Resume Control Register 516
Offset: 0x00022228

A.4 NAND Flash Registers 517

Table 290: Data Flash Control Register (NDCR) 518
Offset: 0x000D0000
Table 291: NAND Interface Timing Parameter 0 Register (NDTR0CS0) 524
Offset: 0x000D0004
Table 292: NAND Interface Timing Parameter 1 Register (NDTR1CS0) 527
Offset: 0x000D000C
Table 293: NAND Controller Status Register (NDSR) 529
Offset: 0x000D0014
Table 294: NAND Controller Page Count Register (NDPCR) 534
Offset: 0x000D0018
Table 295: NAND Controller Bad Block Registers (NDBBRx) 535
Offset: 0x000D001C
Table 296: NAND ECC Control (NDECCCTRL) Register 535
Offset: 0x000D0028
Table 297: NAND Busy Length Count (NDBZCNT) Register 536
Offset: 0x000D002C
Table 298: NAND Controller Data Buffer (NDDB) Register 537
Offset: 0x000D0040
Table 299: NAND Controller Command Buffer 0 (NDCB0) Register 538
Offset: 0x000D0048
Table 300: NAND Controller Command Buffer 1 (NDCB1) Register 541
Offset: 0x000D004C
Table 301: NAND Controller Command Buffer 2 (NDCB2) Register 541
Offset: 0x000D0050
Table 302: NAND Controller Command Buffer 3 (NDCB3) Register 543
Offset: 0x000D0054

Copyright © 2015 Marvell

A.5 Device Bus Controller Registers 544

Table 304: DEV_BOOTCSn Read Parameters Register 544
Offset: 0x00010400

Table 305: DEV_BOOTCSn Write Parameters Register 545
Offset: 0x00010404

Table 306: DEV_CSn[n] Read Parameters Register (n=0–3) 546
Offset: CS0: 0x00010408, CS1: 0x00010410, CS2: 0x00010418, CS3: 0x00010420

Table 307: DEV_CSn[n] Write Parameters Register (n=0–3) 546
Offset: CS0: 0x0001040C, CS1: 0x00010414, CS2: 0x0001041C, CS3: 0x00010424

Table 308: Device Bus Interface Control Register 547
Offset: 0x000104C0

Table 309: Device Bus Error Address Register 548
Offset: 0x000104C4

Table 310: Device Bus Synchronous Control Register 548
Offset: 0x000104C8

Table 311: Device Bus Interrupt Cause Register 550
Offset: 0x000104D0

Table 312: Device Bus Interrupt Mask Register 550
Offset: 0x000104D4

Table 313: Device Bus Miscellaneous Timing Parameters Register 550
Offset: 0x000104D8

A.6 Time Division Multiplexing Registers 552

Table 315: TDM Interface Pin Multiplexing Register 553
Offset: 0x000B4000

Table 316: TDM-Mbus Configuration Register 553
Offset: 0x000B4010

Table 317: Window0 Control Register 554
Offset: 0x000B4030

Table 318: Window0 Base Register 554
Offset: 0x000B4034

Table 319: Window1 Control Register 554
Offset: 0x000B4040

Table 320: Window1 Base Register 555
Offset: 0x000B4044

Table 321: Window2 Control Register 555
Offset: 0x000B4050

Table 322: Window2 Base Register 555
Offset: 0x000B4054

Table 323: Window3 Control Register 556
Offset: 0x000B4060

Table 324: Window3 Base Register 556
Offset: 0x000B4064

Table 325: TDM-Mbus Configuration 1 Register 556
Offset: 0x000B4070

Table 326: PCM Control Register 557
Offset: 0x000B0000

Document Classification: Proprietary Information

Table 327: Channel Time Slot Control Register 559
Offset: 0x000B0004
Table 328: Channel 0 Delay Control Register 560
Offset: 0x000B0008
Table 329: Channel 1 Delay Control Register 560
Offset: 0x000B000C
Table 330: Channel n Enable and Disable Register (n=0–1) 561
Offset: Channel0: 0x000B0010, Channel1: 0x000B0020
Table 331: Channel n Buffer Ownership Register (n=0–1) 561
Offset: Channel0: 0x000B0014, Channel1: 0x000B0024
Table 332: Channel 0 Transmit Data Start Address Register 562
Offset: 0x000B0018
Table 333: Channel 0 Receive Data Start Address Register 562
Offset: 0x000B001C
Table 334: Channel 1 Transmit Data Start Address Register 562
Offset: 0x000B0028
Table 335: Channel 1 Receive Data Start Address Register 562
Offset: 0x000B002C
Table 336: Channel n Total Sample Count Register (n=0–1) 563
Offset: Channel0: 0x000B0030, Channel1: 0x000B0034
Table 337: Number of Time Slots Register 563
Offset: 0x000B0038
Table 338: TDM PCM Clock Rate Divider Register 564
Offset: 0x000B003C
Table 339: Interrupt Event Mask Register 564
Offset: 0x000B0040
Table 340: Interrupt Status Mask Register 564
Offset: 0x000B0048
Table 341: Interrupt Reset Selection Register 565
Offset: 0x000B004C
Table 342: Interrupt Status Register 565
Offset: 0x000B0050
Table 343: Dummy Data for Dummy RX Write Register 566
Offset: 0x000B0054
Table 344: Miscellaneous Control Register 566
Offset: 0x000B0058
Table 345: Channel n Transmit Data Current Address Register (n=0–1) 567
Offset: Channel0: 0x000B0060, Channel1: 0x000B0068
Table 346: Channel n Receive Data Current Address Register (n=0–1) 567
Offset: Channel0: 0x000B0064, Channel1: 0x000B006C
Table 347: Current Time Slot Register 567
Offset: 0x000B0070
Table 348: TDM Revision Register 567
Offset: 0x000B0074
Table 349: TDM Channel n Debug Register (n=0–1) 567
Offset: Channel0: 0x000B0078, Channel1: 0x000B007C
Table 350: TDM DMA Abort Register 1 568
Offset: 0x000B0080

Copyright © 2015 Marvell

Table 351: TDM DMA Abort Register 2 568
Offset: 0x000B0084
Table 352: TDM Channel 0 Wideband Delay Control Register 569
Offset: 0x000B0088
Table 353: TDM Channel 1 Wideband Delay Control Register 569
Offset: 0x000B008C

A.7 PCI Express 2.0 Port 0 Registers 571

Table 355: PCI Express Window0 Control Register 574
Offset: 0x00081820
Table 356: PCI Express Window0 Base Register 574
Offset: 0x00081824
Table 357: PCI Express Window0 Remap Register 575
Offset: 0x0008182C
Table 358: PCI Express Window1 Control Register 575
Offset: 0x00081830
Table 359: PCI Express Window1 Base Register 575
Offset: 0x00081834
Table 360: PCI Express Window1 Remap Register 576
Offset: 0x0008183C
Table 361: PCI Express Window2 Control Register 576
Offset: 0x00081840
Table 362: PCI Express Window2 Base Register 577
Offset: 0x00081844
Table 363: PCI Express Window2 Remap Register 577
Offset: 0x0008184C
Table 364: PCI Express Window3 Control Register 577
Offset: 0x00081850
Table 365: PCI Express Window3 Base Register 578
Offset: 0x00081854
Table 366: PCI Express Window3 Remap Register 578
Offset: 0x0008185C
Table 367: PCI Express Window4 Control Register 578
Offset: 0x00081860
Table 368: PCI Express Window4 Base Register 579
Offset: 0x00081864
Table 369: PCI Express Window4 Remap Register 579
Offset: 0x0008186C
Table 370: PCI Express Window4 Remap (High) Register 579
Offset: 0x00081870
Table 371: PCI Express Window5 Control Register 579
Offset: 0x00081880
Table 372: PCI Express Window5 Base Register 580
Offset: 0x00081884
Table 373: PCI Express Window5 Remap Register 580
Offset: 0x0008188C
Table 374: PCI Express Window5 Remap (High) Register 581
Offset: 0x00081890

Copyright © 2015 Marvell

Table 375: PCI Express Default Window Control Register 581
Offset: 0x000818B0
Table 376: PCI Express Expansion ROM Window Control Register 581
Offset: 0x000818C0
Table 377: PCI Express Expansion ROM Window Remap Register 582
Offset: 0x000818C4
Table 378: PCI Express Window<n+6> Control Register (n=0–9) 582
Offset: x10: 0x00081C00, x11: 0x00081C10...x19: 0x00081C90
Table 379: PCI Express Window<n+6> Base Register (n=0–9) 583
Offset: x10: 0x00081C04, x11: 0x00081C14...x19: 0x00081C94
Table 380: PCI Express Window<n+6> Remap Register (n=0–9) 583
Offset: x10: 0x00081C08, x11: 0x00081C18...x19: 0x00081C98
Table 381: PCI Express Window<n+6> Remap (High) Register (n=0–9) 583
Offset: x10: 0x00081C0C, x11: 0x00081C1C...x19: 0x00081C9C
Table 382: PCI Express BAR1 Control Register 583
Offset: 0x00081804
Table 383: PCI Express BAR2 Control Register 584
Offset: 0x00081808
Table 384: PCI Express Expansion ROM BAR Control Register 584
Offset: 0x0008180C
Table 385: PCI Express Configuration Address Register 585
Offset: 0x000818F8
Table 386: PCI Express Configuration Data Register 585
Offset: 0x000818FC
Table 387: PCI Express Device and Vendor ID Register 585
Offset: 0x00080000
Table 388: PCI Express Command and Status Register 586
Offset: 0x00080004
Table 389: PCI Express Class Code and Revision ID Register 588
Offset: 0x00080008
Table 390: PCI Express BIST Header Type and Cache Line Size Register 588
Offset: 0x0008000C
Table 391: PCI Express BAR0 Internal Register 588
Offset: 0x00080010
Table 392: PCI Express BAR0 Internal (High) Register 589
Offset: 0x00080014
Table 393: PCI Express BAR1 Register 589
Offset: 0x00080018
Table 394: PCI Express BAR1 (High) Register 590
Offset: 0x0008001C
Table 395: PCI Express BAR2 Register 590
Offset: 0x00080020
Table 396: PCI Express BAR2 (High) Register 590
Offset: 0x00080024
Table 397: PCI Express Subsystem Device and Vendor ID Register 590
Offset: 0x0008002C
Table 398: PCI Express Expansion ROM BAR Register 591
Offset: 0x00080030

Copyright © 2015 Marvell

Table 399: PCI Express Capability List Pointer Register 591
Offset: 0x00080034
Table 400: PCI Express Interrupt Pin and Line Register 591
Offset: 0x0008003C
Table 401: PCI Express Power Management Capability Header Register 592
Offset: 0x00080040
Table 402: PCI Express Power Management Control and Status Register 592
Offset: 0x00080044
Table 403: PCI Express MSI Message Control Register 594
Offset: 0x00080050
Table 404: PCI Express MSI Message Address Register 594
Offset: 0x00080054
Table 405: PCI Express MSI Message Address (High) Register 594
Offset: 0x00080058
Table 406: PCI Express MSI Message Data Register 595
Offset: 0x0008005C
Table 407: PCI Express Capability Register 595
Offset: 0x00080060
Table 408: PCI Express Device Capabilities Register 595
Offset: 0x00080064
Table 409: PCI Express Device Control Status Register 597
Offset: 0x00080068
Table 410: PCI Express Link Capabilities Register 599
Offset: 0x0008006C
Table 411: PCI Express Link Control Status Register 601
Offset: 0x00080070
Table 412: PCI Express Device Capabilities 2 Register 603
Offset: 0x00080084
Table 413: PCI Express Device Control Status 2 Register 603
Offset: 0x00080088
Table 414: PCI Express Link Control Status 2 Register 604
Offset: 0x00080090
Table 415: PCI Express Advanced Error Report Header Register 606
Offset: 0x00080100
Table 416: PCI Express Uncorrectable Error Status Register 607
Offset: 0x00080104
Table 417: PCI Express Uncorrectable Error Mask Register 607
Offset: 0x00080108
Table 418: PCI Express Uncorrectable Error Severity Register 608
Offset: 0x0008010C
Table 419: PCI Express Correctable Error Status Register 609
Offset: 0x00080110
Table 420: PCI Express Correctable Error Mask Register 610
Offset: 0x00080114
Table 421: PCI Express Advanced Error Capability and Control Register 611
Offset: 0x00080118
Table 422: PCI Express Header Log First DWORD Register 611
Offset: 0x0008011C

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 423: PCI Express Header Log Second DWORD Register 611
Offset: 0x00080120
Table 424: PCI Express Header Log Third DWORD Register 612
Offset: 0x00080124
Table 425: PCI Express Header Log Fourth DWORD Register 612
Offset: 0x00080128
Table 426: PCI Express Root Error Command Register 612
Offset: 0x0008012C
Table 427: PCI Express Root Error Status Register 612
Offset: 0x00080130
Table 428: PCI Express Error Source Identification Register 613
Offset: 0x00080134
Table 429: PCI Express Control Register 613
Offset: 0x00081A00
Table 430: PCI Express Status Register 615
Offset: 0x00081A04
Table 431: PCI Express Spare Register 616
Offset: 0x00081A08
Table 432: PCI Express Root Complex Set Slot Power Limit (SSPL) Register 616
Offset: 0x00081A0C
Table 433: PCI Express Completion Timeout Register 617
Offset: 0x00081A10
Table 434: PCI Express Root Complex Power Management Event (PME) Register 617
Offset: 0x00081A14
Table 435: PCI Express Power Management Register 618
Offset: 0x00081A18
Table 436: PCI Express Flow Control Register 619
Offset: 0x00081A20
Table 437: PCI Express Replay Timeout (TO) Timer Register 619
Offset: 0x00081A24
Table 438: PCI Express Ack Latency Timer Register 619
Offset: 0x00081A28
Table 439: PCI Express Secondary Bus Register 620
Offset: 0x00081A2C
Table 440: PCI Express Debug Control Register 620
Offset: 0x00081A60
Table 441: PCI Express Debug MAC Status 1 Register 621
Offset: 0x00081A84
Table 442: PCI Express TL Control Register 621
Offset: 0x00081AB0
Table 443: PCI Express Interrupt Cause Register 622
Offset: 0x00081900
Table 444: PCI Express Interrupt Mask Register 625
Offset: 0x00081910
Table 445: PCI Express Mbus Adapter Control Register 625
Offset: 0x000818D0
Table 446: PCI Express Mbus Arbiter Control Register (Low) 626
Offset: 0x000818E0

Copyright © 2015 Marvell

Table 447: PCI Express Mbus Arbiter Control Register (High) 627
Offset: 0x000818E4
Table 448: PCI Express Mbus Arbiter Timeout Register 627
Offset: 0x000818E8

A.8 PCI Express 2.0 Ports 1-3 Registers 628

Table 450: PCI Express Configuration Address Register (p=1–3) 634
Offset: pcie_port_num1: 0x000418F8, pcie_port_num2: 0x000458F8, pcie_port_num3: 0x000498F8
Table 451: PCI Express Configuration Data Register (p=1–3) 634
Offset: pcie_port_num1: 0x000418FC, pcie_port_num2: 0x000458FC, pcie_port_num3: 0x000498FC
Table 452: PCI Express Device and Vendor ID Register (p=1–3) 634
Offset: pcie_port_num1: 0x00040000, pcie_port_num2: 0x00044000, pcie_port_num3: 0x00048000
Table 453: PCI Express Command and Status Register (p=1–3) 635
Offset: pcie_port_num1: 0x00040004, pcie_port_num2: 0x00044004, pcie_port_num3: 0x00048004
Table 454: PCI Express Class Code and Revision ID Register (p=1–3) 637
Offset: pcie_port_num1: 0x00040008, pcie_port_num2: 0x00044008, pcie_port_num3: 0x00048008
Table 455: PCI Express BIST Header Type and Cache Line Size Register (p=1–3) 637
Offset: pcie_port_num1: 0x0004000C, pcie_port_num2: 0x0004400C, pcie_port_num3: 0x0004800C
Table 456: PCI Express BAR0 Internal Register (p=1–3) 637
Offset: pcie_port_num1: 0x00040010, pcie_port_num2: 0x00044010, pcie_port_num3: 0x00048010
Table 457: PCI Express BAR0 Internal (High) Register (p=1–3) 638
Offset: pcie_port_num1: 0x00040014, pcie_port_num2: 0x00044014, pcie_port_num3: 0x00048014
Table 458: PCI Express BAR1 Register (p=1–3) 638
Offset: pcie_port_num1: 0x00040018, pcie_port_num2: 0x00044018, pcie_port_num3: 0x00048018
Table 459: PCI Express BAR1 (High) Register (p=1–3) 639
Offset: pcie_port_num1: 0x0004001C, pcie_port_num2: 0x0004401C, pcie_port_num3: 0x0004801C
Table 460: PCI Express BAR2 Register (p=1–3) 639
Offset: pcie_port_num1: 0x00040020, pcie_port_num2: 0x00044020, pcie_port_num3: 0x00048020
Table 461: PCI Express BAR2 (High) Register (p=1–3) 639
Offset: pcie_port_num1: 0x00040024, pcie_port_num2: 0x00044024, pcie_port_num3: 0x00048024
Table 462: PCI Express Subsystem Device and Vendor ID Register (p=1–3) 639
Offset: pcie_port_num1: 0x0004002C, pcie_port_num2: 0x0004402C, pcie_port_num3: 0x0004802C
Table 463: PCI Express Expansion ROM BAR Register (p=1–3) 640
Offset: pcie_port_num1: 0x00040030, pcie_port_num2: 0x00044030, pcie_port_num3: 0x00048030
Table 464: PCI Express Capability List Pointer Register (p=1–3) 640
Offset: pcie_port_num1: 0x00040034, pcie_port_num2: 0x00044034, pcie_port_num3: 0x00048034
Table 465: PCI Express Interrupt Pin and Line Register (p=1–3) 641
Offset: pcie_port_num1: 0x0004003C, pcie_port_num2: 0x0004403C, pcie_port_num3: 0x0004803C
Table 466: PCI Express Power Management Capability Header Register (p=1–3) 641
Offset: pcie_port_num1: 0x00040040, pcie_port_num2: 0x00044040, pcie_port_num3: 0x00048040
Table 467: PCI Express Power Management Control and Status Register (p=1–3) 642
Offset: pcie_port_num1: 0x00040044, pcie_port_num2: 0x00044044, pcie_port_num3: 0x00048044
Table 468: PCI Express MSI Message Control Register (p=1–3) 643
Offset: pcie_port_num1: 0x00040050, pcie_port_num2: 0x00044050, pcie_port_num3: 0x00048050
Table 469: PCI Express MSI Message Address Register (p=1–3) 643
Offset: pcie_port_num1: 0x00040054, pcie_port_num2: 0x00044054, pcie_port_num3: 0x00048054
Table 470: PCI Express MSI Message Address (High) Register (p=1–3) 644
Offset: pcie_port_num1: 0x00040058, pcie_port_num2: 0x00044058, pcie_port_num3: 0x00048058

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 471: PCI Express MSI Message Data Register (p=1–3) 644
Offset: pcie_port_num1: 0x0004005C, pcie_port_num2: 0x0004405C, pcie_port_num3: 0x0004805C
Table 472: PCI Express Capability Register (p=1–3) 644
Offset: pcie_port_num1: 0x00040060, pcie_port_num2: 0x00044060, pcie_port_num3: 0x00048060
Table 473: PCI Express Device Capabilities Register (p=1–3) 645
Offset: pcie_port_num1: 0x00040064, pcie_port_num2: 0x00044064, pcie_port_num3: 0x00048064
Table 474: PCI Express Device Control Status Register (p=1–3) 646
Offset: pcie_port_num1: 0x00040068, pcie_port_num2: 0x00044068, pcie_port_num3: 0x00048068
Table 475: PCI Express Link Capabilities Register (p=1–3) 648
Offset: pcie_port_num1: 0x0004006C, pcie_port_num2: 0x0004406C, pcie_port_num3: 0x0004806C
Table 476: PCI Express Link Control Status Register (p=1–3) 650
Offset: pcie_port_num1: 0x00040070, pcie_port_num2: 0x00044070, pcie_port_num3: 0x00048070
Table 477: PCI Express Device Capabilities 2 Register (p=1–3) 652
Offset: pcie_port_num1: 0x00040084, pcie_port_num2: 0x00044084, pcie_port_num3: 0x00048084
Table 478: PCI Express Device Control Status 2 Register (p=1–3) 653
Offset: pcie_port_num1: 0x00040088, pcie_port_num2: 0x00044088, pcie_port_num3: 0x00048088
Table 479: PCI Express Link Control Status 2 Register (p=1–3) 653
Offset: pcie_port_num1: 0x00040090, pcie_port_num2: 0x00044090, pcie_port_num3: 0x00048090
Table 480: PCI Express Advanced Error Report Header Register (p=1–3) 656
Offset: pcie_port_num1: 0x00040100, pcie_port_num2: 0x00044100, pcie_port_num3: 0x00048100
Table 481: PCI Express Uncorrectable Error Status Register (p=1–3) 656
Offset: pcie_port_num1: 0x00040104, pcie_port_num2: 0x00044104, pcie_port_num3: 0x00048104
Table 482: PCI Express Uncorrectable Error Mask Register (p=1–3) 657
Offset: pcie_port_num1: 0x00040108, pcie_port_num2: 0x00044108, pcie_port_num3: 0x00048108
Table 483: PCI Express Uncorrectable Error Severity Register (p=1–3) 658
Offset: pcie_port_num1: 0x0004010C, pcie_port_num2: 0x0004410C, pcie_port_num3: 0x0004810C
Table 484: PCI Express Correctable Error Status Register (p=1–3) 659
Offset: pcie_port_num1: 0x00040110, pcie_port_num2: 0x00044110, pcie_port_num3: 0x00048110
Table 485: PCI Express Correctable Error Mask Register (p=1–3) 660
Offset: pcie_port_num1: 0x00040114, pcie_port_num2: 0x00044114, pcie_port_num3: 0x00048114
Table 486: PCI Express Advanced Error Capability and Control Register (p=1–3) 660
Offset: pcie_port_num1: 0x00040118, pcie_port_num2: 0x00044118, pcie_port_num3: 0x00048118
Table 487: PCI Express Header Log First DWORD Register (p=1–3) 661
Offset: pcie_port_num1: 0x0004011C, pcie_port_num2: 0x0004411C, pcie_port_num3: 0x0004811C
Table 488: PCI Express Header Log Second DWORD Register (p=1–3) 661
Offset: pcie_port_num1: 0x00040120, pcie_port_num2: 0x00044120, pcie_port_num3: 0x00048120
Table 489: PCI Express Header Log Third DWORD Register (p=1–3) 662
Offset: pcie_port_num1: 0x00040124, pcie_port_num2: 0x00044124, pcie_port_num3: 0x00048124
Table 490: PCI Express Header Log Fourth DWORD Register (p=1–3) 662
Offset: pcie_port_num1: 0x00040128, pcie_port_num2: 0x00044128, pcie_port_num3: 0x00048128
Table 491: PCI Express Root Error Command Register (p=1–3) 662
Offset: pcie_port_num1: 0x0004012C, pcie_port_num2: 0x0004412C, pcie_port_num3: 0x0004812C
Table 492: PCI Express Root Error Status Register (p=1–3) 663
Offset: pcie_port_num1: 0x00040130, pcie_port_num2: 0x00044130, pcie_port_num3: 0x00048130
Table 493: PCI Express Error Source Identification Register (p=1–3) 663
Offset: pcie_port_num1: 0x00040134, pcie_port_num2: 0x00044134, pcie_port_num3: 0x00048134
Table 494: PCI Express Window0 Control Register (p=1–3) 664
Offset: pcie_port_num1: 0x00041820, pcie_port_num2: 0x00045820, pcie_port_num3: 0x00049820

Copyright © 2015 Marvell

Table 495: PCI Express Window0 Base Register (p=1–3) 664
Offset: pcie_port_num1: 0x00041824, pcie_port_num2: 0x00045824, pcie_port_num3: 0x00049824
Table 496: PCI Express Window0 Remap Register (p=1–3) 665
Offset: pcie_port_num1: 0x0004182C, pcie_port_num2: 0x0004582C, pcie_port_num3: 0x0004982C
Table 497: PCI Express Window1 Control Register (p=1–3) 665
Offset: pcie_port_num1: 0x00041830, pcie_port_num2: 0x00045830, pcie_port_num3: 0x00049830
Table 498: PCI Express Window1 Base Register (p=1–3) 666
Offset: pcie_port_num1: 0x00041834, pcie_port_num2: 0x00045834, pcie_port_num3: 0x00049834
Table 499: PCI Express Window1 Remap Register (p=1–3) 666
Offset: pcie_port_num1: 0x0004183C, pcie_port_num2: 0x0004583C, pcie_port_num3: 0x0004983C
Table 500: PCI Express Window2 Control Register (p=1–3) 666
Offset: pcie_port_num1: 0x00041840, pcie_port_num2: 0x00045840, pcie_port_num3: 0x00049840
Table 501: PCI Express Window2 Base Register (p=1–3) 667
Offset: pcie_port_num1: 0x00041844, pcie_port_num2: 0x00045844, pcie_port_num3: 0x00049844
Table 502: PCI Express Window2 Remap Register (p=1–3) 667
Offset: pcie_port_num1: 0x0004184C, pcie_port_num2: 0x0004584C, pcie_port_num3: 0x0004984C
Table 503: PCI Express Window3 Control Register (p=1–3) 667
Offset: pcie_port_num1: 0x00041850, pcie_port_num2: 0x00045850, pcie_port_num3: 0x00049850
Table 504: PCI Express Window3 Base Register (p=1–3) 668
Offset: pcie_port_num1: 0x00041854, pcie_port_num2: 0x00045854, pcie_port_num3: 0x00049854
Table 505: PCI Express Window3 Remap Register (p=1–3) 668
Offset: pcie_port_num1: 0x0004185C, pcie_port_num2: 0x0004585C, pcie_port_num3: 0x0004985C
Table 506: PCI Express Window4 Control Register (p=1–3) 669
Offset: pcie_port_num1: 0x00041860, pcie_port_num2: 0x00045860, pcie_port_num3: 0x00049860
Table 507: PCI Express Window4 Base Register (p=1–3) 669
Offset: pcie_port_num1: 0x00041864, pcie_port_num2: 0x00045864, pcie_port_num3: 0x00049864
Table 508: PCI Express Window4 Remap Register (p=1–3) 669
Offset: pcie_port_num1: 0x0004186C, pcie_port_num2: 0x0004586C, pcie_port_num3: 0x0004986C
Table 509: PCI Express Window4 Remap (High) Register (p=1–3) 670
Offset: pcie_port_num1: 0x00041870, pcie_port_num2: 0x00045870, pcie_port_num3: 0x00049870
Table 510: PCI Express Window5 Control Register (p=1–3) 670
Offset: pcie_port_num1: 0x00041880, pcie_port_num2: 0x00045880, pcie_port_num3: 0x00049880
Table 511: PCI Express Window5 Base Register (p=1–3) 671
Offset: pcie_port_num1: 0x00041884, pcie_port_num2: 0x00045884, pcie_port_num3: 0x00049884
Table 512: PCI Express Window5 Remap Register (p=1–3) 671
Offset: pcie_port_num1: 0x0004188C, pcie_port_num2: 0x0004588C, pcie_port_num3: 0x0004988C
Table 513: PCI Express Window5 Remap (High) Register (p=1–3) 671
Offset: pcie_port_num1: 0x00041890, pcie_port_num2: 0x00045890, pcie_port_num3: 0x00049890
Table 514: PCI Express Default Window Control Register (p=1–3) 671
Offset: pcie_port_num1: 0x000418B0, pcie_port_num2: 0x000458B0, pcie_port_num3: 0x000498B0
Table 515: PCI Express Expansion ROM Window Control Register (p=1–3) 672
Offset: pcie_port_num1: 0x000418C0, pcie_port_num2: 0x000458C0, pcie_port_num3: 0x000498C0
Table 516: PCI Express Expansion ROM Window Remap Register (p=1–3) 672
Offset: pcie_port_num1: 0x000418C4, pcie_port_num2: 0x000458C4, pcie_port_num3: 0x000498C4
Table 517: PCI Express Window<n+6> Control Register (n=0–9, p=1–3) 673
Offset: pcie_port_num1x10: 0x00041C00, pcie_port_num1x11: 0x00041C10...pcie_port_num3x19: 0x00049C90

Document Classification: Proprietary Information

Table 518: PCI Express Window<n+6> Base Register (n=0–9, p=1–3) 673
Offset: pcie_port_num1x10: 0x00041C04, pcie_port_num1x11: 0x00041C14...pcie_port_num3x19: 0x00049C94
Table 519: PCI Express Window<n+6> Remap Register (n=0–9, p=1–3) 674
Offset: pcie_port_num1x10: 0x00041C08, pcie_port_num1x11: 0x00041C18...pcie_port_num3x19: 0x00049C98
Table 520: PCI Express Window<n+6> Remap (High) Register (n=0–9, p=1–3) 674
Offset: pcie_port_num1x10: 0x00041C0C, pcie_port_num1x11: 0x00041C1C...pcie_port_num3x19: 0x00049C9C
Table 521: PCI Express Mbus Adapter Control Register (p=1–3) 674
Offset: pcie_port_num1: 0x000418D0, pcie_port_num2: 0x000458D0, pcie_port_num3: 0x000498D0
Table 522: PCI Express Mbus Arbiter Control Register (Low) (p=1–3) 675
Offset: pcie_port_num1: 0x000418E0, pcie_port_num2: 0x000458E0, pcie_port_num3: 0x000498E0
Table 523: PCI Express Mbus Arbiter Control Register (High) (p=1–3) 675
Offset: pcie_port_num1: 0x000418E4, pcie_port_num2: 0x000458E4, pcie_port_num3: 0x000498E4
Table 524: PCI Express Mbus Arbiter Timeout Register (p=1–3) 676
Offset: pcie_port_num1: 0x000418E8, pcie_port_num2: 0x000458E8, pcie_port_num3: 0x000498E8
Table 525: PCI Express Control Register (p=1–3) 676
Offset: pcie_port_num1: 0x00041A00, pcie_port_num2: 0x00045A00, pcie_port_num3: 0x00049A00
Table 526: PCI Express Status Register (p=1–3) 678
Offset: pcie_port_num1: 0x00041A04, pcie_port_num2: 0x00045A04, pcie_port_num3: 0x00049A04
Table 527: PCI Express Spare Register (p=1–3) 679
Offset: pcie_port_num1: 0x00041A08, pcie_port_num2: 0x00045A08, pcie_port_num3: 0x00049A08
Table 528: PCI Express Root Complex Set Slot Power Limit (SSPL) Register (p=1–3) 679
Offset: pcie_port_num1: 0x00041A0C, pcie_port_num2: 0x00045A0C, pcie_port_num3: 0x00049A0C
Table 529: PCI Express Completion Timeout Register (p=1–3) 680
Offset: pcie_port_num1: 0x00041A10, pcie_port_num2: 0x00045A10, pcie_port_num3: 0x00049A10
Table 530: PCI Express Root Complex Power Management Event (PME) Register (p=1–3) 680
Offset: pcie_port_num1: 0x00041A14, pcie_port_num2: 0x00045A14, pcie_port_num3: 0x00049A14
Table 531: PCI Express Power Management Register (p=1–3) 681
Offset: pcie_port_num1: 0x00041A18, pcie_port_num2: 0x00045A18, pcie_port_num3: 0x00049A18
Table 532: PCI Express Flow Control Register (p=1–3) 682
Offset: pcie_port_num1: 0x00041A20, pcie_port_num2: 0x00045A20, pcie_port_num3: 0x00049A20
Table 533: PCI Express Replay Timeout (TO) Timer Register (p=1–3) 682
Offset: pcie_port_num1: 0x00041A24, pcie_port_num2: 0x00045A24, pcie_port_num3: 0x00049A24
Table 534: PCI Express Ack Latency Timer Register (p=1–3) 682
Offset: pcie_port_num1: 0x00041A28, pcie_port_num2: 0x00045A28, pcie_port_num3: 0x00049A28
Table 535: PCI Express Secondary Bus Register (p=1–3) 683
Offset: pcie_port_num1: 0x00041A2C, pcie_port_num2: 0x00045A2C, pcie_port_num3: 0x00049A2C
Table 536: PCI Express BAR1 Control Register (p=1–3) 683
Offset: pcie_port_num1: 0x00041804, pcie_port_num2: 0x00045804, pcie_port_num3: 0x00049804
Table 537: PCI Express BAR2 Control Register (p=1–3) 684
Offset: pcie_port_num1: 0x00041808, pcie_port_num2: 0x00045808, pcie_port_num3: 0x00049808
Table 538: PCI Express Expansion ROM BAR Control Register (p=1–3) 684
Offset: pcie_port_num1: 0x0004180C, pcie_port_num2: 0x0004580C, pcie_port_num3: 0x0004980C
Table 539: PCI Express Interrupt Cause Register (p=1–3) 684
Offset: pcie_port_num1: 0x00041900, pcie_port_num2: 0x00045900, pcie_port_num3: 0x00049900
Table 540: PCI Express Interrupt Mask Register (p=1–3) 687
Offset: pcie_port_num1: 0x00041910, pcie_port_num2: 0x00045910, pcie_port_num3: 0x00049910

Copyright © 2015 Marvell

Table 541: PCI Express Debug Control Register (p=1–3) 688
Offset: pcie_port_num1: 0x00041A60, pcie_port_num2: 0x00045A60, pcie_port_num3: 0x00049A60
Table 542: PCI Express Debug MAC Status 1 Register (p=1–3) 688
Offset: pcie_port_num1: 0x00041A84, pcie_port_num2: 0x00045A84, pcie_port_num3: 0x00049A84
Table 543: PCI Express TL Control Register (p=1–3) 689
Offset: pcie_port_num1: 0x00041AB0, pcie_port_num2: 0x00045AB0, pcie_port_num3: 0x00049AB0

A.9 USB 2.0 Registers 691

Table 545: USB 2.0 Bridge Interrupt Cause Register 691
Offset: 0x00058310
Table 546: USB 2.0 Bridge Interrupt Mask Register 692
Offset: 0x00058314
Table 547: USB 2.0 Core Interrupt Mask Register 692
Offset: 0x00058318
Table 548: USB 2.0 Bridge Error Address Register 692
Offset: 0x0005831C
Table 549: USB 2.0 Bridge Control Register 693
Offset: 0x00058300
Table 550: USB 2.0 Window0 Control Register 693
Offset: 0x00058320
Table 551: USB 2.0 Window0 Base Register 694
Offset: 0x00058324
Table 552: USB 2.0 Window1 Control Register 694
Offset: 0x00058330
Table 553: USB 2.0 Window1 Base Register 695
Offset: 0x00058334
Table 554: USB 2.0 Window2 Control Register 695
Offset: 0x00058340
Table 555: USB 2.0 Window2 Base Register 695
Offset: 0x00058344
Table 556: USB 2.0 Window3 Control Register 696
Offset: 0x00058350
Table 557: USB 2.0 Window3 Base Register 696
Offset: 0x00058354
Table 558: USB 2.0 Power Control Register 696
Offset: 0x00058400
Table 559: USB 2.0 PHY Suspend/Resume Control Register 697
Offset: 0x00058404

A.10 USB2 PHY (UTMI PHY) Registers 698

Table 561: PLL Control 0 Register (n=0–2) 699
Offset: Port0: 0x000C0000, Port1: 0x000C2000, Port2: 0x000C4000
Table 562: PLL Control 1 Register (n=0–2) 700
Offset: Port0: 0x000C0004, Port1: 0x000C2004, Port2: 0x000C4004
Table 563: Calibration Control Register (n=0–2) 701
Offset: Port0: 0x000C0008, Port1: 0x000C2008, Port2: 0x000C4008
Table 564: TX Channel Control Register 0 (n=0–2) 703
Offset: Port0: 0x000C000C, Port1: 0x000C200C, Port2: 0x000C400C

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 565: TX Channel Control Register 1 (n=0–2) 705
Offset: Port0: 0x000C0010, Port1: 0x000C2010, Port2: 0x000C4010
Table 566: Rx Channel Control 0 Register (n=0–2) 705
Offset: Port0: 0x000C0014, Port1: 0x000C2014, Port2: 0x000C4014
Table 567: Rx Channel Control 1 Register (n=0–2) 707
Offset: Port0: 0x000C0018, Port1: 0x000C2018, Port2: 0x000C4018
Table 568: Digital Control Register 0 (n=0–2) 708
Offset: Port0: 0x000C001C, Port1: 0x000C201C, Port2: 0x000C401C
Table 569: Digital Control Register 1 (n=0–2) 710
Offset: Port0: 0x000C0020, Port1: 0x000C2020, Port2: 0x000C4020
Table 570: Test Control and Status Register 0 (n=0–2) 712
Offset: Port0: 0x000C0024, Port1: 0x000C2024, Port2: 0x000C4024
Table 571: Test Control and Status Register 1 (n=0–2) 714
Offset: Port0: 0x000C0028, Port1: 0x000C2028, Port2: 0x000C4028
Table 572: Monitor Register (n=0–2) 714
Offset: Port0: 0x000C002C, Port1: 0x000C202C, Port2: 0x000C402C
Table 573: PHY RESERVE Register (n=0–2) 714
Offset: Port0: 0x000C0030, Port1: 0x000C2030, Port2: 0x000C4030
Table 574: PHY Charger Detector Control Register (n=0–2) 715
Offset: Port0: 0x000C0038, Port1: 0x000C2038, Port2: 0x000C4038
Table 575: Reserved. Register (n=0–2) 716
Offset: Port0: 0x000C003C, Port1: 0x000C203C, Port2: 0x000C403C

A.11 USB 3.0 Host Controller Registers 717

Table 577: CAPLENGTH Register (n=0–1) 720
Offset: usb host port0: 0x000F0000, usb host port1: 0x000F8000
Table 578: HCIVERSION Register (n=0–1) 720
Offset: usb host port0: 0x000F0002, usb host port1: 0x000F8002
Table 579: HCSPARAMS_1 Register (n=0–1) 721
Offset: usb host port0: 0x000F0004, usb host port1: 0x000F8004
Table 580: HCSPARAMS_2 Register (n=0–1) 721
Offset: usb host port0: 0x000F0008, usb host port1: 0x000F8008
Table 581: HCSPARAMS_3 Register (n=0–1) 722
Offset: usb host port0: 0x000F000C, usb host port1: 0x000F800C
Table 582: HCCPARAMS Register (n=0–1) 722
Offset: usb host port0: 0x000F0010, usb host port1: 0x000F8010
Table 583: DBOFF Register (n=0–1) 723
Offset: usb host port0: 0x000F0014, usb host port1: 0x000F8014
Table 584: RTSOFF Register (n=0–1) 723
Offset: usb host port0: 0x000F0018, usb host port1: 0x000F8018
Table 585: USBCMD Register (n=0–1) 724
Offset: usb host port0: 0x000F0020, usb host port1: 0x000F8020
Table 586: USBSTS Register (n=0–1) 724
Offset: usb host port0: 0x000F0024, usb host port1: 0x000F8024
Table 587: PAGESIZE Register (n=0–1) 725
Offset: usb host port0: 0x000F0028, usb host port1: 0x000F8028
Table 588: DNCTRL Register (n=0–1) 725
Offset: usb host port0: 0x000F0034, usb host port1: 0x000F8034

Table 589: CRCR Register (n=0–1) 726
Offset: usb host port0: 0x000F0038, usb host port1: 0x000F8038
Table 590: DCBAAP Register (n=0–1) 727
Offset: usb host port0: 0x000F0050, usb host port1: 0x000F8050
Table 591: xHCI_CONFIG Register (n=0–1) 727
Offset: usb host port0: 0x000F0058, usb host port1: 0x000F8058
Table 592: USB3_PORTSC Register (n=0–1) 727
Offset: usb host port0: 0x000F0420, usb host port1: 0x000F8420
Table 593: USB3_PORTPMSC Register (n=0–1) 729
Offset: usb host port0: 0x000F0424, usb host port1: 0x000F8424
Table 594: USB3_PORTLI Register (n=0–1) 729
Offset: usb host port0: 0x000F0428, usb host port1: 0x000F8428
Table 595: USB3_PORTHLPMC Register (n=0–1) 729
Offset: usb host port0: 0x000F042C, usb host port1: 0x000F842C
Table 596: USB2_PORTSC Register (n=0–1) 730
Offset: usb host port0: 0x000F0430, usb host port1: 0x000F8430
Table 597: USB2_PORTPMSC Register (n=0–1) 731
Offset: usb host port0: 0x000F0434, usb host port1: 0x000F8434
Table 598: USB_PORTLI Register (n=0–1) 731
Offset: usb host port0: 0x000F0438, usb host port1: 0x000F8438
Table 599: USB2_PORTHLPMC Register (n=0–1) 732
Offset: usb host port0: 0x000F043C, usb host port1: 0x000F843C
Table 600: MFINDEX Register (n=0–1) 732
Offset: usb host port0: 0x000F1000, usb host port1: 0x000F9000
Table 601: IMAN Register (n=0–1) 732
Offset: usb host port0: 0x000F1020, usb host port1: 0x000F9020
Table 602: IMOD Register (n=0–1) 733
Offset: usb host port0: 0x000F1024, usb host port1: 0x000F9024
Table 603: ERSTSZ Register (n=0–1) 733
Offset: usb host port0: 0x000F1028, usb host port1: 0x000F9028
Table 604: ERSTBA Register (n=0–1) 733
Offset: usb host port0: 0x000F1030, usb host port1: 0x000F9030
Table 605: ERDP Register (n=0–1) 733
Offset: usb host port0: 0x000F1038, usb host port1: 0x000F9038
Table 606: CMD_DB Register (n=0–1) 734
Offset: usb host port0: 0x000F2000, usb host port1: 0x000FA000
Table 607: DB1 Register (n=0–1) 734
Offset: usb host port0: 0x000F2004, usb host port1: 0x000FA004
Table 608: DB2 Register (n=0–1) 734
Offset: usb host port0: 0x000F2008, usb host port1: 0x000FA008
Table 609: DB3 Register (n=0–1) 735
Offset: usb host port0: 0x000F200C, usb host port1: 0x000FA00C
Table 610: DB4 Register (n=0–1) 735
Offset: usb host port0: 0x000F2010, usb host port1: 0x000FA010
Table 611: DB5 Register (n=0–1) 735
Offset: usb host port0: 0x000F2014, usb host port1: 0x000FA014
Table 612: DB6 Register (n=0–1) 735
Offset: usb host port0: 0x000F2018, usb host port1: 0x000FA018

Copyright © 2015 Marvell

Table 613: DB7 Register (n=0–1) 736
Offset: usb host port0: 0x000F201C, usb host port1: 0x000FA01C
Table 614: DB8 Register (n=0–1) 736
Offset: usb host port0: 0x000F2020, usb host port1: 0x000FA020
Table 615: DB9 Register (n=0–1) 736
Offset: usb host port0: 0x000F2024, usb host port1: 0x000FA024
Table 616: DB10 Register (n=0–1) 736
Offset: usb host port0: 0x000F2028, usb host port1: 0x000FA028
Table 617: DB11 Register (n=0–1) 737
Offset: usb host port0: 0x000F202C, usb host port1: 0x000FA02C
Table 618: DB12 Register (n=0–1) 737
Offset: usb host port0: 0x000F2030, usb host port1: 0x000FA030
Table 619: DB13 Register (n=0–1) 737
Offset: usb host port0: 0x000F2034, usb host port1: 0x000FA034
Table 620: DB14 Register (n=0–1) 737
Offset: usb host port0: 0x000F2038, usb host port1: 0x000FA038
Table 621: DB15 Register (n=0–1) 738
Offset: usb host port0: 0x000F203C, usb host port1: 0x000FA03C
Table 622: DB16 Register (n=0–1) 738
Offset: usb host port0: 0x000F2040, usb host port1: 0x000FA040
Table 623: DB17 Register (n=0–1) 738
Offset: usb host port0: 0x000F2044, usb host port1: 0x000FA044
Table 624: DB18 Register (n=0–1) 738
Offset: usb host port0: 0x000F2048, usb host port1: 0x000FA048
Table 625: DB19 Register (n=0–1) 739
Offset: usb host port0: 0x000F204C, usb host port1: 0x000FA04C
Table 626: DB20 Register (n=0–1) 739
Offset: usb host port0: 0x000F2050, usb host port1: 0x000FA050
Table 627: DB21 Register (n=0–1) 739
Offset: usb host port0: 0x000F2054, usb host port1: 0x000FA054
Table 628: DB22 Register (n=0–1) 739
Offset: usb host port0: 0x000F2058, usb host port1: 0x000FA058
Table 629: DB23 Register (n=0–1) 740
Offset: usb host port0: 0x000F205C, usb host port1: 0x000FA05C
Table 630: DB24 Register (n=0–1) 740
Offset: usb host port0: 0x000F2060, usb host port1: 0x000FA060
Table 631: DB25 Register (n=0–1) 740
Offset: usb host port0: 0x000F2064, usb host port1: 0x000FA064
Table 632: DB26 Register (n=0–1) 740
Offset: usb host port0: 0x000F2068, usb host port1: 0x000FA068
Table 633: DB27 Register (n=0–1) 741
Offset: usb host port0: 0x000F206C, usb host port1: 0x000FA06C
Table 634: DB28 Register (n=0–1) 741
Offset: usb host port0: 0x000F2070, usb host port1: 0x000FA070
Table 635: DB29 Register (n=0–1) 741
Offset: usb host port0: 0x000F2074, usb host port1: 0x000FA074
Table 636: DB30 Register (n=0–1) 741
Offset: usb host port0: 0x000F2078, usb host port1: 0x000FA078

Copyright © 2015 Marvell

Table 637: DB31 Register (n=0–1) 742
Offset: usb host port0: 0x000F207C, usb host port1: 0x000FA07C
Table 638: xECP_SUP_CAP_USB2_R1 Register (n=0–1) 742
Offset: usb host port0: 0x000F2800, usb host port1: 0x000FA800
Table 639: xECP_SUP_CAP_USB2_R2 Register (n=0–1) 742
Offset: usb host port0: 0x000F2804, usb host port1: 0x000FA804
Table 640: xECP_SUP_CAP_USB2_R3 Register (n=0–1) 742
Offset: usb host port0: 0x000F2808, usb host port1: 0x000FA808
Table 641: xECP_SUP_CAP_USB3_R1 Register (n=0–1) 743
Offset: usb host port0: 0x000F2810, usb host port1: 0x000FA810
Table 642: xECP_SUP_CAP_USB3_R2 Register (n=0–1) 743
Offset: usb host port0: 0x000F2814, usb host port1: 0x000FA814
Table 643: xECP_SUP_CAP_USB3_R3 Register (n=0–1) 744
Offset: usb host port0: 0x000F2818, usb host port1: 0x000FA818

A.12 USB 3.0 Host Controller Bridge Registers 745

Table 645: Bridge Window Control Register (n=0–1, N=0–7) 745
Offset: usb host port0Windows0: 0x000F4000, usb host port0Windows1: 0x000F4008...usb host port1Windows7: 0x000FC038
Table 646: Bridge Window Base Register (n=0–1, N=0–7) 746
Offset: usb host port0Windows0: 0x000F4004, usb host port0Windows1: 0x000F400C...usb host port1Windows7: 0x000FC03C
Table 647: Mbus Control Register (n=0–1) 746
Offset: usb host port0: 0x000F4040, usb host port1: 0x000FC040
Table 648: Sync Barrier WindowN Control Register (n=0–1, N=0–1) 747
Offset: usb host port0SyncbWin0: 0x000F4044, usb host port0SyncbWin1: 0x000F404C, usb host port1SyncbWin0: 0x000FC044, usb host port1SyncbWin1: 0x000FC04C
Table 649: Sync Barrier WindowN Base Register (n=0–1, N=0–1) 747
Offset: usb host port0SyncbWin0: 0x000F4048, usb host port0SyncbWin1: 0x000F4050, usb host port1SyncbWin0: 0x000FC048, usb host port1SyncbWin1: 0x000FC050

A.13 USB 3.0 Device Registers 748

Table 651: TSEQ_COUNTER_U1U2_TIMEOUT_REG Register 763
Offset: 0x00050190
Table 652: TIMER_TIMEOUT_REG_1 Register 764
Offset: 0x00050194
Table 653: TIMER_TIMEOUT_REG_2 Register 764
Offset: 0x00050198
Table 654: LFPS_TX_CONFIG_REG Register 764
Offset: 0x0005019C
Table 655: LFPS_RX_CONFIG_REG Register 765
Offset: 0x000501A0
Table 656: LFPS_WR_TRESET_VAL_REG Register 765
Offset: 0x000501A4
Table 657: COUNTER_DELAY_TX_REG Register 766
Offset: 0x000501A8
Table 658: PHY_CONFIG_REG Register 766
Offset: 0x000501AC

Copyright © 2015 Marvell

Table 659: LOOP_BACK_REG Register 767
Offset: 0x000501B0
Table 660: BERT_INTERVAL_REG Register 767
Offset: 0x000501B4
Table 661: SHALLOW_LOOP_BACK_REG Register 768
Offset: 0x000501B8
Table 662: POWER_MANAGEMENT_DEVICE_REG Register 768
Offset: 0x000500C8
Table 663: POWER_MANAGEMENT_SOC_REG Register 769
Offset: 0x000500CC
Table 664: LOW_POWER_STATUS_REG Register 770
Offset: 0x000500D0
Table 665: SOFTWARE_RESET_REG Register 770
Offset: 0x000500D4
Table 666: TOP_INTERRUPT_STATUS_REG Register 771
Offset: 0x000500D8
Table 667: TOP_INTERRUPT_ENABLE_REG Register 772
Offset: 0x000500DC
Table 668: PME_INTERRUPT_STATUS_EN_REG Register 772
Offset: 0x000500E0
Table 669: AHB_TIMEOUT_REG Register 773
Offset: 0x000500E4
Table 670: DSP_DISCONNECT Register 773
Offset: 0x00058400
Table 671: CONFIGURE_FOR_HS Register 774
Offset: 0x00058401
Table 672: BURST_CLOSURE Register 774
Offset: 0x00058402
Table 673: DISABLE_SCRAMBLING Register 775
Offset: 0x00058403
Table 674: DISABLE_STALL_IN_U0 Register 776
Offset: 0x00058404
Table 675: DISABLE_LUP_LDN Register 776
Offset: 0x00058405
Table 676: TEST_MODE Register 777
Offset: 0x000584FF
Table 677: PAIR_CAPABILITY Register 778
Offset: 0x00058500
Table 678: LOOPBACK_EN Register 778
Offset: 0x00058E00
Table 679: RX_BURST_COUNT_0 Register 779
Offset: 0x00058E01
Table 680: RX_BURST_COUNT_1 Register 780
Offset: 0x00058E02
Table 681: RX_BURST_COUNT_2 Register 780
Offset: 0x00058E03
Table 682: RX_BURST_COUNT_3 Register 781
Offset: 0x00058E04

Copyright © 2015 Marvell

Table 683: RX_ERR_COUNT_0 Register 781
Offset: 0x00058E05
Table 684: RX_ERR_COUNT_1 Register 782
Offset: 0x00058E06
Table 685: RX_ERR_COUNT_2 Register 782
Offset: 0x00058E07
Table 686: RX_ERR_COUNT_3 Register 783
Offset: 0x00058E08
Table 687: RX_COUNT_RESET Register 783
Offset: 0x00058E09
Table 688: TX_COMP_MODE_EN Register 784
Offset: 0x00058E0A
Table 689: ANALOG_LOOPBACK_EN Register 784
Offset: 0x00058E0B
Table 690: PM_CLOCK_GATE_EN Register 785
Offset: 0x00058F00
Table 691: Timer_Hibern8_Enter_TX Register 785
Offset: 0x00058F02
Table 692: Timer_Hibern8_Residency Register 786
Offset: 0x00058F04
Table 693: Num_of_Cycles_for_1us Register 786
Offset: 0x00058F06
Table 694: MPHY_RMMI_Register_Base_Addr 786
Offset: 0x00058F08
Table 695: Num_of_Cycles_for_10us Register 787
Offset: 0x00058F0A
Table 696: SSIC_Dev_Timer_Tretrain Register 787
Offset: 0x00058F0E
Table 697: SSIC_Dev_Pollingactive_Residency Register 787
Offset: 0x00058F10
Table 698: SSIC_Control_Status_Register 788
Offset: 0x00058F20
Table 699: SSIC_Status_Interrupt_Enable_Register 789
Offset: 0x00058F22
Table 700: Timer_Tx_Tactive_Time Register 790
Offset: 0x00058F28
Table 701: Timer_Rx_Tactive_Time Register 790
Offset: 0x00058F2A
Table 702: DEV_INFO_REG Register 791
Offset: 0x00050000
Table 703: DEV_TAG_REG Register 791
Offset: 0x00050004
Table 704: LOW_POWER_LFPS_SIGNAL_VAL_REG_1 Register 792
Offset: 0x00050008
Table 705: LOW_POWER_LFPS_SIGNAL_VAL_REG_2 Register 792
Offset: 0x0005000C
Table 706: LOW_POWER_LFPS_SIGNAL_VAL_REG_3 Register 793
Offset: 0x00050010

Copyright © 2015 Marvell

Table 707: LOW_POWER_LFPS_SIGNAL_VAL_REG_4 Register 793
Offset: 0x00050014
Table 708: LOW_POWER_LFPS_SIGNAL_VAL_REG_5 Register 793
Offset: 0x00050018
Table 709: LOW_POWER_LFPS_SIGNAL_VAL_REG_6 Register 794
Offset: 0x0005001C
Table 710: COUNTER_PULSE_VALUE_REG Register 794
Offset: 0x00050020
Table 711: REF_INT_STATUS_REG Register 794
Offset: 0x00050024
Table 712: REF_INT_ENABLE_REG Register 795
Offset: 0x00050028
Table 713: USB3_GLOBAL_CONTROL_STATUS_REG Register 796
Offset: 0x0005002C
Table 714: SYSTEM_DEBUG_REG Register 797
Offset: 0x00050030
Table 715: EP0_ACK_TIMEOUT_VALUE_REG Register 799
Offset: 0x00050034
Table 716: ACK_TIMEOUT_VALUE_REG Register 800
Offset: 0x00050038
Table 717: RX_IDLE_DELAY_COUNTER_REG Register 800
Offset: 0x0005003C
Table 718: COUNTER_PULSE_VALUE_REG_1 Register 800
Offset: 0x00050040
Table 719: TEST_BUS_MUX_REG Register 801
Offset: 0x00050044
Table 720: CLOCK_GATING_ENABLE_REG Register 801
Offset: 0x00050048
Table 721: POWER_GATING_ENABLE_REG Register 802
Offset: 0x0005004C
Table 722: DEV_USB_ADDRESS_REG Register 803
Offset: 0x00050320
Table 723: FUNCTION_WAKEUP_REG Register 803
Offset: 0x00050324
Table 724: ENDPOINT_0_CONFIG_REG Register 803
Offset: 0x00050328
Table 725: OUT_ENDPOINT_CONFIG_REG_1 Register 804
Offset: 0x0005032C
Table 726: OUT_ENDPOINT_CONFIG_REG_2 Register 806
Offset: 0x00050330
Table 727: OUT_ENDPOINT_CONFIG_REG_3 Register 807
Offset: 0x00050334
Table 728: OUT_ENDPOINT_CONFIG_REG_4 Register 808
Offset: 0x00050338
Table 729: OUT_ENDPOINT_CONFIG_REG_5 Register 810
Offset: 0x0005033C
Table 730: OUT_ENDPOINT_CONFIG_REG_6 Register 811
Offset: 0x00050340

Copyright © 2015 Marvell

Table 731: OUT_ENDPOINT_CONFIG_REG_7 Register 813
Offset: 0x00050344
Table 732: OUT_ENDPOINT_CONFIG_REG_8 Register 814
Offset: 0x00050348
Table 733: OUT_ENDPOINT_CONFIG_REG_9 Register 816
Offset: 0x0005034C
Table 734: OUT_ENDPOINT_CONFIG_REG_10 Register 817
Offset: 0x00050350
Table 735: OUT_ENDPOINT_CONFIG_REG_11 Register 819
Offset: 0x00050354
Table 736: OUT_ENDPOINT_CONFIG_REG_12 Register 820
Offset: 0x00050358
Table 737: OUT_ENDPOINT_CONFIG_REG_13 Register 822
Offset: 0x0005035C
Table 738: OUT_ENDPOINT_CONFIG_REG_14 Register 823
Offset: 0x00050360
Table 739: OUT_ENDPOINT_CONFIG_REG_15 Register 825
Offset: 0x00050364
Table 740: IN_ENDPOINT_CONFIG_REG_1 Register 826
Offset: 0x00050368
Table 741: IN_ENDPOINT_CONFIG_REG_2 Register 828
Offset: 0x0005036C
Table 742: IN_ENDPOINT_CONFIG_REG_3 Register 829
Offset: 0x00050370
Table 743: IN_ENDPOINT_CONFIG_REG_4 Register 831
Offset: 0x00050374
Table 744: IN_ENDPOINT_CONFIG_REG_5 Register 832
Offset: 0x00050378
Table 745: IN_ENDPOINT_CONFIG_REG_6 Register 834
Offset: 0x0005037C
Table 746: IN_ENDPOINT_CONFIG_REG_7 Register 835
Offset: 0x00050380
Table 747: IN_ENDPOINT_CONFIG_REG_8 Register 837
Offset: 0x00050384
Table 748: IN_ENDPOINT_CONFIG_REG_9 Register 838
Offset: 0x00050388
Table 749: IN_ENDPOINT_CONFIG_REG_10 Register 840
Offset: 0x0005038C
Table 750: IN_ENDPOINT_CONFIG_REG_11 Register 841
Offset: 0x00050390
Table 751: IN_ENDPOINT_CONFIG_REG_12 Register 843
Offset: 0x00050394
Table 752: IN_ENDPOINT_CONFIG_REG_13 Register 844
Offset: 0x00050398
Table 753: IN_ENDPOINT_CONFIG_REG_14 Register 846
Offset: 0x0005039C
Table 754: IN_ENDPOINT_CONFIG_REG_15 Register 847
Offset: 0x000503A0

Copyright © 2015 Marvell

Table 755: PHY_CFG_REG Register 849
Offset: 0x000503A4
Table 756: BUS_INTERVAL_LTM_REG Register 851
Offset: 0x000503A8
Table 757: ISO_TIME_STAMP_COUNTER_REG Register 851
Offset: 0x000503AC
Table 758: CREDIT_HP_AND_U2_TIMER_REG Register 851
Offset: 0x000503B0
Table 759: PM_LC_TIMER_TIME_OUT_REG Register 852
Offset: 0x000503B4
Table 760: SS_CORE_INTERRUPT_REG Register 852
Offset: 0x000503B8
Table 761: SS_CORE_INTERRUPT_EN_REG Register 854
Offset: 0x000503BC
Table 762: EP_IN_BINTERVAL_REG_1_2_3 Register 855
Offset: 0x000503C0
Table 763: EP_IN_BINTERVAL_REG_4_5_6_7 Register 855
Offset: 0x000503C4
Table 764: EP_IN_BINTERVAL_REG_8_9_10_11 Register 855
Offset: 0x000503C8
Table 765: EP_IN_BINTERVAL_REG_12_13_14_15 Register 856
Offset: 0x000503CC
Table 766: EP_OUT_BINTERVAL_REG_1_2_3 Register 856
Offset: 0x000503D0
Table 767: EP_OUT_BINTERVAL_REG_4_5_6_7 Register 857
Offset: 0x000503D4
Table 768: EP_OUT_BINTERVAL_REG_8_9_10_11 Register 857
Offset: 0x000503D8
Table 769: EP_OUT_BINTERVAL_REG_12_13_14_15 Register 858
Offset: 0x000503DC
Table 770: EP_HOST_ERROR_STATUS_REG Register 858
Offset: 0x000503E0
Table 771: UX_EXIT_DETECTION_DELAY_REG Register 859
Offset: 0x000503E8
Table 772: TX_TSI_NUM_REG Register 860
Offset: 0x000503EC
Table 773: START_STATE_DELAY_AND_MIN_TS2_HOT_RESET_REG Register 860
Offset: 0x000503F0
Table 774: LOWPOWER_MANAGEMENT_REG Register 861
Offset: 0x000503F4
Table 775: USB3_SETUP_DP_LOW Register 861
Offset: 0x000503F8
Table 776: USB3_SETUP_DP_HIGH_REG Register 861
Offset: 0x000503FC
Table 777: USB3_SETUP_CONTROL_REG Register 861
Offset: 0x00050400
Table 778: DELAY_U1_CNT_INIT_VALUE_REG Register 862
Offset: 0x00050404

Copyright © 2015 Marvell

Table 779: DELAY_U2_CNT_INIT_VALUE_REG Register 862
Offset: 0x00050408
Table 780: NO_BURST_MODE_REG Register 862
Offset: 0x0005040C
Table 781: DMA_GLOBAL_CONFIG_REG Register 862
Offset: 0x000507D0
Table 782: BULK_STREAMING_ENABLE_REG Register 863
Offset: 0x000507D4
Table 783: EP_OUT_RCV_STREAM_ID_REG_1 Register 865
Offset: 0x000507D8
Table 784: EP_OUT_RCV_STREAM_ID_REG_2 Register 865
Offset: 0x000507DC
Table 785: EP_OUT_RCV_STREAM_ID_REG_3 Register 866
Offset: 0x000507E0
Table 786: EP_OUT_RCV_STREAM_ID_REG_4 Register 866
Offset: 0x000507E4
Table 787: EP_OUT_RCV_STREAM_ID_REG_5 Register 866
Offset: 0x000507E8
Table 788: EP_OUT_RCV_STREAM_ID_REG_6 Register 866
Offset: 0x000507EC
Table 789: EP_OUT_RCV_STREAM_ID_REG_7 Register 867
Offset: 0x000507F0
Table 790: EP_OUT_RCV_STREAM_ID_REG_8 Register 867
Offset: 0x000507F4
Table 791: EP_OUT_RCV_STREAM_ID_REG_9 Register 867
Offset: 0x000507F8
Table 792: EP_OUT_RCV_STREAM_ID_REG_10 Register 867
Offset: 0x000507FC
Table 793: EP_OUT_RCV_STREAM_ID_REG_11 Register 868
Offset: 0x00050800
Table 794: EP_OUT_RCV_STREAM_ID_REG_12 Register 868
Offset: 0x00050804
Table 795: EP_OUT_RCV_STREAM_ID_REG_13 Register 868
Offset: 0x00050808
Table 796: EP_OUT_RCV_STREAM_ID_REG_14 Register 868
Offset: 0x0005080C
Table 797: EP_OUT_RCV_STREAM_ID_REG_15 Register 869
Offset: 0x00050810
Table 798: EP_IN_RCV_STREAM_ID_REG_1 Register 869
Offset: 0x00050814
Table 799: EP_IN_RCV_STREAM_ID_REG_2 Register 869
Offset: 0x00050818
Table 800: EP_IN_RCV_STREAM_ID_REG_3 Register 869
Offset: 0x0005081C
Table 801: EP_IN_RCV_STREAM_ID_REG_4 Register 870
Offset: 0x00050820
Table 802: EP_IN_RCV_STREAM_ID_REG_5 Register 870
Offset: 0x00050824

Document Classification: Proprietary Information

Table 803: EP_IN_RCV_STREAM_ID_REG_6 Register 870
Offset: 0x00050828
Table 804: EP_IN_RCV_STREAM_ID_REG_7 Register 870
Offset: 0x0005082C
Table 805: EP_IN_RCV_STREAM_ID_REG_8 Register 871
Offset: 0x00050830
Table 806: EP_IN_RCV_STREAM_ID_REG_9 Register 871
Offset: 0x00050834
Table 807: EP_IN_RCV_STREAM_ID_REG_10 Register 871
Offset: 0x00050838
Table 808: EP_IN_RCV_STREAM_ID_REG_11 Register 871
Offset: 0x0005083C
Table 809: EP_IN_RCV_STREAM_ID_REG_12 Register 872
Offset: 0x00050840
Table 810: EP_IN_RCV_STREAM_ID_REG_13 Register 872
Offset: 0x00050844
Table 811: EP_IN_RCV_STREAM_ID_REG_14 Register 872
Offset: 0x00050848
Table 812: EP_IN_RCV_STREAM_ID_REG_15 Register 872
Offset: 0x0005084C
Table 813: DMA_COMPLETE_SUCCESS_STATUS Register 873
Offset: 0x00050850
Table 814: DMA_COMPLETE_ERROR_STATUS Register 878
Offset: 0x00050854
Table 815: DMA_BD_FETCH_ERROR_STATUS_REG Register 883
Offset: 0x00050858
Table 816: DMA_BD_FETCH_ERROR_EN_REG Register 885
Offset: 0x0005085C
Table 817: DMA_DATA_ERROR_STATUS_REG Register 886
Offset: 0x00050860
Table 818: DMA_DATA_ERROR_EN_REG Register 888
Offset: 0x00050864
Table 819: DMA_ERROR_HANDLING_MODE_REG Register 890
Offset: 0x00050868
Table 820: EP_OUT_RX_DMA_CONFIG_REG_0 Register 890
Offset: 0x0005086C
Table 821: EP_OUT_RX_DMA_CONFIG_REG_1 Register 891
Offset: 0x00050870
Table 822: EP_OUT_RX_DMA_CONFIG_REG_2 Register 891
Offset: 0x00050874
Table 823: EP_OUT_RX_DMA_CONFIG_REG_3 Register 892
Offset: 0x00050878
Table 824: EP_OUT_RX_DMA_CONFIG_REG_4 Register 892
Offset: 0x0005087C
Table 825: EP_OUT_RX_DMA_CONFIG_REG_5 Register 893
Offset: 0x00050880
Table 826: EP_OUT_RX_DMA_CONFIG_REG_6 Register 893
Offset: 0x00050884

Table 827: EP_OUT_RX_DMA_CONFIG_REG_7 Register 894
Offset: 0x00050888
Table 828: EP_OUT_RX_DMA_CONFIG_REG_8 Register 894
Offset: 0x0005088C
Table 829: EP_OUT_RX_DMA_CONFIG_REG_9 Register 895
Offset: 0x00050890
Table 830: EP_OUT_RX_DMA_CONFIG_REG_10 Register 895
Offset: 0x00050894
Table 831: EP_OUT_RX_DMA_CONFIG_REG_11 Register 896
Offset: 0x00050898
Table 832: EP_OUT_RX_DMA_CONFIG_REG_12 Register 896
Offset: 0x0005089C
Table 833: EP_OUT_RX_DMA_CONFIG_REG_13 Register 897
Offset: 0x000508A0
Table 834: EP_OUT_RX_DMA_CONFIG_REG_14 Register 897
Offset: 0x000508A4
Table 835: EP_OUT_RX_DMA_CONFIG_REG_15 Register 898
Offset: 0x000508A8
Table 836: EP_IN_TX_DMA_CONFIG_REG_0 Register 898
Offset: 0x000508AC
Table 837: EP_IN_TX_DMA_CONFIG_REG_1 Register 899
Offset: 0x000508B0
Table 838: EP_IN_TX_DMA_CONFIG_REG_2 Register 899
Offset: 0x000508B4
Table 839: EP_IN_TX_DMA_CONFIG_REG_3 Register 900
Offset: 0x000508B8
Table 840: EP_IN_TX_DMA_CONFIG_REG_4 Register 900
Offset: 0x000508BC
Table 841: EP_IN_TX_DMA_CONFIG_REG_5 Register 901
Offset: 0x000508C0
Table 842: EP_IN_TX_DMA_CONFIG_REG_6 Register 901
Offset: 0x000508C4
Table 843: EP_IN_TX_DMA_CONFIG_REG_7 Register 902
Offset: 0x000508C8
Table 844: EP_IN_TX_DMA_CONFIG_REG_8 Register 902
Offset: 0x000508CC
Table 845: EP_IN_TX_DMA_CONFIG_REG_9 Register 903
Offset: 0x000508D0
Table 846: EP_IN_TX_DMA_CONFIG_REG_10 Register 903
Offset: 0x000508D4
Table 847: EP_IN_TX_DMA_CONFIG_REG_11 Register 904
Offset: 0x000508D8
Table 848: EP_IN_TX_DMA_CONFIG_REG_12 Register 904
Offset: 0x000508DC
Table 849: EP_IN_TX_DMA_CONFIG_REG_13 Register 905
Offset: 0x000508E0
Table 850: EP_IN_TX_DMA_CONFIG_REG_14 Register 905
Offset: 0x000508E4

Document Classification: Proprietary Information

Table 851: EP_IN_TX_DMA_CONFIG_REG_15 Register 906
Offset: 0x000508E8
Table 852: EP_OUT_RX_DMA_START_ADDRESS_REG_0 Register 906
Offset: 0x000508EC
Table 853: EP_OUT_RX_DMA_START_ADDRESS_REG_1 Register 907
Offset: 0x000508F0
Table 854: EP_OUT_RX_DMA_START_ADDRESS_REG_2 Register 907
Offset: 0x000508F4
Table 855: EP_OUT_RX_DMA_START_ADDRESS_REG_3 Register 907
Offset: 0x000508F8
Table 856: EP_OUT_RX_DMA_START_ADDRESS_REG_4 Register 907
Offset: 0x000508FC
Table 857: EP_OUT_RX_DMA_START_ADDRESS_REG_5 Register 907
Offset: 0x00050900
Table 858: EP_OUT_RX_DMA_START_ADDRESS_REG_6 Register 908
Offset: 0x00050904
Table 859: EP_OUT_RX_DMA_START_ADDRESS_REG_7 Register 908
Offset: 0x00050908
Table 860: EP_OUT_RX_DMA_START_ADDRESS_REG_8 Register 908
Offset: 0x0005090C
Table 861: EP_OUT_RX_DMA_START_ADDRESS_REG_9 Register 908
Offset: 0x00050910
Table 862: EP_OUT_RX_DMA_START_ADDRESS_REG_10 Register 908
Offset: 0x00050914
Table 863: EP_OUT_RX_DMA_START_ADDRESS_REG_11 Register 909
Offset: 0x00050918
Table 864: EP_OUT_RX_DMA_START_ADDRESS_REG_12 Register 909
Offset: 0x0005091C
Table 865: EP_OUT_RX_DMA_START_ADDRESS_REG_13 Register 909
Offset: 0x00050920
Table 866: EP_OUT_RX_DMA_START_ADDRESS_REG_14 Register 909
Offset: 0x00050924
Table 867: EP_OUT_RX_DMA_START_ADDRESS_REG_15 Register 909
Offset: 0x00050928
Table 868: EP_IN_TX_DMA_START_ADDRESS_REG_0 Register 910
Offset: 0x0005092C
Table 869: EP_IN_TX_DMA_START_ADDRESS_REG_1 Register 910
Offset: 0x00050930
Table 870: EP_IN_TX_DMA_START_ADDRESS_REG_2 Register 910
Offset: 0x00050934
Table 871: EP_IN_TX_DMA_START_ADDRESS_REG_3 Register 910
Offset: 0x00050938
Table 872: EP_IN_TX_DMA_START_ADDRESS_REG_4 Register 910
Offset: 0x0005093C
Table 873: EP_IN_TX_DMA_START_ADDRESS_REG_5 Register 911
Offset: 0x00050940
Table 874: EP_IN_TX_DMA_START_ADDRESS_REG_6 Register 911
Offset: 0x00050944

Copyright © 2015 Marvell

Table 875: EP_IN_TX_DMA_START_ADDRESS_REG_7 Register 911
Offset: 0x00050948
Table 876: EP_IN_TX_DMA_START_ADDRESS_REG_8 Register 911
Offset: 0x0005094C
Table 877: EP_IN_TX_DMA_START_ADDRESS_REG_9 Register 911
Offset: 0x00050950
Table 878: EP_IN_TX_DMA_START_ADDRESS_REG_10 Register 912
Offset: 0x00050954
Table 879: EP_IN_TX_DMA_START_ADDRESS_REG_11 Register 912
Offset: 0x00050958
Table 880: EP_IN_TX_DMA_START_ADDRESS_REG_12 Register 912
Offset: 0x0005095C
Table 881: EP_IN_TX_DMA_START_ADDRESS_REG_13 Register 912
Offset: 0x00050960
Table 882: EP_IN_TX_DMA_START_ADDRESS_REG_14 Register 912
Offset: 0x00050964
Table 883: EP_IN_TX_DMA_START_ADDRESS_REG_15 Register 913
Offset: 0x00050968
Table 884: EP_OUT_DMA_ERROR_ADDRESS_REG_0 Register 913
Offset: 0x0005096C
Table 885: EP_OUT_DMA_ERROR_ADDRESS_REG_1 Register 913
Offset: 0x00050970
Table 886: EP_OUT_DMA_ERROR_ADDRESS_REG_2 Register 913
Offset: 0x00050974
Table 887: EP_OUT_DMA_ERROR_ADDRESS_REG_3 Register 914
Offset: 0x00050978
Table 888: EP_OUT_DMA_ERROR_ADDRESS_REG_4 Register 914
Offset: 0x0005097C
Table 889: EP_OUT_DMA_ERROR_ADDRESS_REG_5 Register 914
Offset: 0x00050980
Table 890: EP_OUT_DMA_ERROR_ADDRESS_REG_6 Register 915
Offset: 0x00050984
Table 891: EP_OUT_DMA_ERROR_ADDRESS_REG_7 Register 915
Offset: 0x00050988
Table 892: EP_OUT_DMA_ERROR_ADDRESS_REG_8 Register 915
Offset: 0x0005098C
Table 893: EP_OUT_DMA_ERROR_ADDRESS_REG_9 Register 916
Offset: 0x00050990
Table 894: EP_OUT_DMA_ERROR_ADDRESS_REG_10 Register 916
Offset: 0x00050994
Table 895: EP_OUT_DMA_ERROR_ADDRESS_REG_11 Register 916
Offset: 0x00050998
Table 896: EP_OUT_DMA_ERROR_ADDRESS_REG_12 Register 917
Offset: 0x0005099C
Table 897: EP_OUT_DMA_ERROR_ADDRESS_REG_13 Register 917
Offset: 0x000509A0
Table 898: EP_OUT_DMA_ERROR_ADDRESS_REG_14 Register 917
Offset: 0x000509A4

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 899: EP_OUT_DMA_ERROR_ADDRESS_REG_15 Register 918
Offset: 0x000509A8
Table 900: EP_IN_DMA_ERROR_ADDRESS_REG_0 Register 918
Offset: 0x000509AC
Table 901: EP_IN_DMA_ERROR_ADDRESS_REG_1 Register 918
Offset: 0x000509B0
Table 902: EP_IN_DMA_ERROR_ADDRESS_REG_2 Register 919
Offset: 0x000509B4
Table 903: EP_IN_DMA_ERROR_ADDRESS_REG_3 Register 919
Offset: 0x000509B8
Table 904: EP_IN_DMA_ERROR_ADDRESS_REG_4 Register 919
Offset: 0x000509BC
Table 905: EP_IN_DMA_ERROR_ADDRESS_REG_5 Register 920
Offset: 0x000509C0
Table 906: EP_IN_DMA_ERROR_ADDRESS_REG_6 Register 920
Offset: 0x000509C4
Table 907: EP_IN_DMA_ERROR_ADDRESS_REG_7 Register 920
Offset: 0x000509C8
Table 908: EP_IN_DMA_ERROR_ADDRESS_REG_8 Register 921
Offset: 0x000509CC
Table 909: EP_IN_DMA_ERROR_ADDRESS_REG_9 Register 921
Offset: 0x000509D0
Table 910: EP_IN_DMA_ERROR_ADDRESS_REG_10 Register 921
Offset: 0x000509D4
Table 911: EP_IN_DMA_ERROR_ADDRESS_REG_11 Register 922
Offset: 0x000509D8
Table 912: EP_IN_DMA_ERROR_ADDRESS_REG_12 Register 922
Offset: 0x000509DC
Table 913: EP_IN_DMA_ERROR_ADDRESS_REG_13 Register 922
Offset: 0x000509E0
Table 914: EP_IN_DMA_ERROR_ADDRESS_REG_14 Register 923
Offset: 0x000509E4
Table 915: EP_IN_DMA_ERROR_ADDRESS_REG_15 Register 923
Offset: 0x000509E8
Table 916: DMA_SUSPEND_REG Register 923
Offset: 0x000509EC
Table 917: DMA_SUSPEND_DONE_REG Register 926
Offset: 0x000509F0
Table 918: DMA_HALT_REG Register 928
Offset: 0x000509F4
Table 919: DMA_HALT_DONE_REG Register 928
Offset: 0x000509F8
Table 920: AXI_TIMEOUT_VAL_REG_0 Register 929
Offset: 0x000509FC
Table 921: AXI_STATE_REG_1 Register 929
Offset: 0x00050A00
Table 922: AXI_STATE_REG_2 Register 930
Offset: 0x00050A04

Copyright © 2015 Marvell

Table 923: AXI_STATE_REG_3 Register 930
Offset: 0x00050A08
Table 924: SS_SYS_INT_REG Register 931
Offset: 0x00050A0C
Table 925: SS_SYS_INT_EN_REG Register 934
Offset: 0x00050A10
Table 926: AXI_ERROR_RESPONSE_CODE_REG_1 Register 935
Offset: 0x00050A14
Table 927: AXI_ERROR_REPONSE_CODE_REG_2 Register 936
Offset: 0x00050A18
Table 928: AXI_ERROR_RESPONSE_CODE_REG_3 Register 936
Offset: 0x00050A1C
Table 929: AXI_ERROR_RESPONSE_CODE_REG_4 Register 937
Offset: 0x00050A20
Table 930: DMA_STATE_REG_1 Register 937
Offset: 0x00050A24
Table 931: DMA_STATE_REG_2 Register 937
Offset: 0x00050A28
Table 932: DMA_STATE_REG_3 Register 938
Offset: 0x00050A2C
Table 933: DMA_STATE_REG_4 Register 938
Offset: 0x00050A30
Table 934: DMA_STATE_REG_5 Register 939
Offset: 0x00050A34
Table 935: SEGMENT_COUNTER_REG_0 Register 939
Offset: 0x00050A38
Table 936: SEGMENT_COUNTER_REG_1 Register 939
Offset: 0x00050A3C
Table 937: SEGMENT_COUNTER_REG_2 Register 940
Offset: 0x00050A40
Table 938: SEGMENT_COUNTER_REG_3 Register 940
Offset: 0x00050A44
Table 939: SEGMENT_COUNTER_REG_4 Register 940
Offset: 0x00050A48
Table 940: SEGMENT_COUNTER_REG_5 Register 941
Offset: 0x00050A4C
Table 941: SEGMENT_COUNTER_REG_6 Register 941
Offset: 0x00050A50
Table 942: SEGMENT_COUNTER_REG_7 Register 941
Offset: 0x00050A54
Table 943: SEGMENT_COUNTER_REG_8 Register 942
Offset: 0x00050A58
Table 944: SEGMENT_COUNTER_REG_9 Register 942
Offset: 0x00050A5C
Table 945: SEGMENT_COUNTER_REG_10 Register 942
Offset: 0x00050A60
Table 946: SEGMENT_COUNTER_REG_11 Register 943
Offset: 0x00050A64

Copyright © 2015 Marvell

Table 947: SEGMENT_COUNTER_REG_12 Register 943
Offset: 0x00050A68
Table 948: SEGMENT_COUNTER_REG_13 Register 943
Offset: 0x00050A6C
Table 949: SEGMENT_COUNTER_REG_14 Register 944
Offset: 0x00050A70
Table 950: SEGMENT_COUNTER_REG_15 Register 944
Offset: 0x00050A74
Table 951: EP_IN_DONEQ_START_ADDRESS_REG_0 Register 944
Offset: 0x00050A78
Table 952: EP_IN_DONEQ_START_ADDRESS_REG_1 Register 944
Offset: 0x00050A7C
Table 953: EP_IN_DONEQ_START_ADDRESS_REG_2 Register 945
Offset: 0x00050A80
Table 954: EP_IN_DONEQ_START_ADDRESS_REG_3 Register 945
Offset: 0x00050A84
Table 955: EP_IN_DONEQ_START_ADDRESS_REG_4 Register 945
Offset: 0x00050A88
Table 956: EP_IN_DONEQ_START_ADDRESS_REG_5 Register 945
Offset: 0x00050A8C
Table 957: EP_IN_DONEQ_START_ADDRESS_REG_6 Register 945
Offset: 0x00050A90
Table 958: EP_IN_DONEQ_START_ADDRESS_REG_7 Register 945
Offset: 0x00050A94
Table 959: EP_IN_DONEQ_START_ADDRESS_REG_8 Register 946
Offset: 0x00050A98
Table 960: EP_IN_DONEQ_START_ADDRESS_REG_9 Register 946
Offset: 0x00050A9C
Table 961: EP_IN_DONEQ_START_ADDRESS_REG_10 Register 946
Offset: 0x00050AA0
Table 962: EP_IN_DONEQ_START_ADDRESS_REG_11 Register 946
Offset: 0x00050AA4
Table 963: EP_IN_DONEQ_START_ADDRESS_REG_12 Register 946
Offset: 0x00050AA8
Table 964: EP_IN_DONEQ_START_ADDRESS_REG_13 Register 946
Offset: 0x00050AAC
Table 965: EP_IN_DONEQ_START_ADDRESS_REG_14 Register 947
Offset: 0x00050AB0
Table 966: EP_IN_DONEQ_START_ADDRESS_REG_15 Register 947
Offset: 0x00050AB4
Table 967: EP_IN_DONEQ_END_ADDRESS_REG_0 Register 947
Offset: 0x00050AB8
Table 968: EP_IN_DONEQ_END_ADDRESS_REG_1 Register 947
Offset: 0x00050ABC
Table 969: EP_IN_DONEQ_END_ADDRESS_REG_2 Register 947
Offset: 0x00050AC0
Table 970: EP_IN_DONEQ_END_ADDRESS_REG_3 Register 947
Offset: 0x00050AC4

Copyright © 2015 Marvell

Table 971: EP_IN_DONEQ_END_ADDRESS_REG_4 Register 948
Offset: 0x00050AC8
Table 972: EP_IN_DONEQ_END_ADDRESS_REG_5 Register 948
Offset: 0x00050ACC
Table 973: EP_IN_DONEQ_END_ADDRESS_REG_6 Register 948
Offset: 0x00050AD0
Table 974: EP_IN_DONEQ_END_ADDRESS_REG_7 Register 948
Offset: 0x00050AD4
Table 975: EP_IN_DONEQ_END_ADDRESS_REG_8 Register 948
Offset: 0x00050AD8
Table 976: EP_IN_DONEQ_END_ADDRESS_REG_9 Register 948
Offset: 0x00050ADC
Table 977: EP_IN_DONEQ_END_ADDRESS_REG_10 Register 949
Offset: 0x00050AE0
Table 978: EP_IN_DONEQ_END_ADDRESS_REG_11 Register 949
Offset: 0x00050AE4
Table 979: EP_IN_DONEQ_END_ADDRESS_REG_12 Register 949
Offset: 0x00050AE8
Table 980: EP_IN_DONEQ_END_ADDRESS_REG_13 Register 949
Offset: 0x00050AEC
Table 981: EP_IN_DONEQ_END_ADDRESS_REG_14 Register 949
Offset: 0x00050AF0
Table 982: EP_IN_DONEQ_END_ADDRESS_REG_15 Register 949
Offset: 0x00050AF4
Table 983: EP_OUT_DONEQ_START_ADDRESS_REG_0 Register 950
Offset: 0x00050AF8
Table 984: EP_OUT_DONEQ_START_ADDRESS_REG_1 Register 950
Offset: 0x00050AFC
Table 985: EP_OUT_DONEQ_START_ADDRESS_REG_2 Register 950
Offset: 0x00050B00
Table 986: EP_OUT_DONEQ_START_ADDRESS_REG_3 Register 950
Offset: 0x00050B04
Table 987: EP_OUT_DONEQ_START_ADDRESS_REG_4 Register 950
Offset: 0x00050B08
Table 988: EP_OUT_DONEQ_START_ADDRESS_REG_5 Register 950
Offset: 0x00050B0C
Table 989: EP_OUT_DONEQ_START_ADDRESS_REG_6 Register 951
Offset: 0x00050B10
Table 990: EP_OUT_DONEQ_START_ADDRESS_REG_7 Register 951
Offset: 0x00050B14
Table 991: EP_OUT_DONEQ_START_ADDRESS_REG_8 Register 951
Offset: 0x00050B18
Table 992: EP_OUT_DONEQ_START_ADDRESS_REG_9 Register 951
Offset: 0x00050B1C
Table 993: EP_OUT_DONEQ_START_ADDRESS_REG_10 Register 951
Offset: 0x00050B20
Table 994: EP_OUT_DONEQ_START_ADDRESS_REG_11 Register 951
Offset: 0x00050B24

Document Classification: Proprietary Information

Table 995: EP_OUT_DONEQ_START_ADDRESS_REG_12 Register 952
Offset: 0x00050B28
Table 996: EP_OUT_DONEQ_START_ADDRESS_REG_13 Register 952
Offset: 0x00050B2C
Table 997: EP_OUT_DONEQ_START_ADDRESS_REG_14 Register 952
Offset: 0x00050B30
Table 998: EP_OUT_DONEQ_START_ADDRESS_REG_15 Register 952
Offset: 0x00050B34
Table 999: EP_OUT_DONEQ_END_ADDRESS_REG_0 Register 952
Offset: 0x00050B38
Table 1000: EP_OUT_DONEQ_END_ADDRESS_REG_1 Register 952
Offset: 0x00050B3C
Table 1001: EP_OUT_DONEQ_END_ADDRESS_REG_2 Register 953
Offset: 0x00050B40
Table 1002: EP_OUT_DONEQ_END_ADDRESS_REG_3 Register 953
Offset: 0x00050B44
Table 1003: EP_OUT_DONEQ_END_ADDRESS_REG_4 Register 953
Offset: 0x00050B48
Table 1004: EP_OUT_DONEQ_END_ADDRESS_REG_5 Register 953
Offset: 0x00050B4C
Table 1005: EP_OUT_DONEQ_END_ADDRESS_REG_6 Register 953
Offset: 0x00050B50
Table 1006: EP_OUT_DONEQ_END_ADDRESS_REG_7 Register 953
Offset: 0x00050B54
Table 1007: EP_OUT_DONEQ_END_ADDRESS_REG_8 Register 954
Offset: 0x00050B58
Table 1008: EP_OUT_DONEQ_END_ADDRESS_REG_9 Register 954
Offset: 0x00050B5C
Table 1009: EP_OUT_DONEQ_END_ADDRESS_REG_10 Register 954
Offset: 0x00050B60
Table 1010: EP_OUT_DONEQ_END_ADDRESS_REG_11 Register 954
Offset: 0x00050B64
Table 1011: EP_OUT_DONEQ_END_ADDRESS_REG_12 Register 954
Offset: 0x00050B68
Table 1012: EP_OUT_DONEQ_END_ADDRESS_REG_13 Register 954
Offset: 0x00050B6C
Table 1013: EP_OUT_DONEQ_END_ADDRESS_REG_14 Register 955
Offset: 0x00050B70
Table 1014: EP_OUT_DONEQ_END_ADDRESS_REG_15 Register 955
Offset: 0x00050B74
Table 1015: EP_IN_DONEQ_WRITE_POINTER_REG_0 Register 955
Offset: 0x00050B78
Table 1016: EP_IN_DONEQ_WRITE_POINTER_REG_1 Register 955
Offset: 0x00050B7C
Table 1017: EP_IN_DONEQ_WRITE_POINTER_REG_2 Register 955
Offset: 0x00050B80
Table 1018: EP_IN_DONEQ_WRITE_POINTER_REG_3 Register 956
Offset: 0x00050B84

Copyright © 2015 Marvell

Table 1019: EP_IN_DONEQ_WRITE_POINTER_REG_4 Register 956
Offset: 0x00050B88
Table 1020: EP_IN_DONEQ_WRITE_POINTER_REG_5 Register 956
Offset: 0x00050B8C
Table 1021: EP_IN_DONEQ_WRITE_POINTER_REG_6 Register 956
Offset: 0x00050B90
Table 1022: EP_IN_DONEQ_WRITE_POINTER_REG_7 Register 956
Offset: 0x00050B94
Table 1023: EP_IN_DONEQ_WRITE_POINTER_REG_8 Register 957
Offset: 0x00050B98
Table 1024: EP_IN_DONEQ_WRITE_POINTER_REG_9 Register 957
Offset: 0x00050B9C
Table 1025: EP_IN_DONEQ_WRITE_POINTER_REG_10 Register 957
Offset: 0x00050BA0
Table 1026: EP_IN_DONEQ_WRITE_POINTER_REG_11 Register 957
Offset: 0x00050BA4
Table 1027: EP_IN_DONEQ_WRITE_POINTER_REG_12 Register 957
Offset: 0x00050BA8
Table 1028: EP_IN_DONEQ_WRITE_POINTER_REG_13 Register 958
Offset: 0x00050BAC
Table 1029: EP_IN_DONEQ_WRITE_POINTER_REG_14 Register 958
Offset: 0x00050BB0
Table 1030: EP_IN_DONEQ_WRITE_POINTER_REG_15 Register 958
Offset: 0x00050BB4
Table 1031: EP_IN_DONEQ_READ_POINTER_REG_0 Register 958
Offset: 0x00050BB8
Table 1032: EP_IN_DONEQ_READ_POINTER_REG_1 Register 958
Offset: 0x00050BBC
Table 1033: EP_IN_DONEQ_READ_POINTER_REG_2 Register 959
Offset: 0x00050BC0
Table 1034: EP_IN_DONEQ_READ_POINTER_REG_3 Register 959
Offset: 0x00050BC4
Table 1035: EP_IN_DONEQ_READ_POINTER_REG_4 Register 959
Offset: 0x00050BC8
Table 1036: EP_IN_DONEQ_READ_POINTER_REG_5 Register 959
Offset: 0x00050BCC
Table 1037: EP_IN_DONEQ_READ_POINTER_REG_6 Register 960
Offset: 0x00050BD0
Table 1038: EP_IN_DONEQ_READ_POINTER_REG_7 Register 960
Offset: 0x00050BD4
Table 1039: EP_IN_DONEQ_READ_POINTER_REG_8 Register 960
Offset: 0x00050BD8
Table 1040: EP_IN_DONEQ_READ_POINTER_REG_9 Register 960
Offset: 0x00050BDC
Table 1041: EP_IN_DONEQ_READ_POINTER_REG_10 Register 961
Offset: 0x00050BE0
Table 1042: EP_IN_DONEQ_READ_POINTER_REG_11 Register 961
Offset: 0x00050BE4

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 1043: EP_IN_DONEQ_READ_POINTER_REG_12 Register 961
Offset: 0x00050BE8
Table 1044: EP_IN_DONEQ_READ_POINTER_REG_13 Register 961
Offset: 0x00050BEC
Table 1045: EP_IN_DONEQ_READ_POINTER_REG_14 Register 962
Offset: 0x00050BF0
Table 1046: EP_IN_DONEQ_READ_POINTER_REG_15 Register 962
Offset: 0x00050BF4
Table 1047: EP_OUT_DONEQ_WRITE_POINTER_REG_0 Register 962
Offset: 0x00050BF8
Table 1048: EP_OUT_DONEQ_WRITE_POINTER_REG_1 Register 962
Offset: 0x00050BFC
Table 1049: EP_OUT_DONEQ_WRITE_POINTER_REG_2 Register 962
Offset: 0x00050C00
Table 1050: EP_OUT_DONEQ_WRITE_POINTER_REG_3 Register 963
Offset: 0x00050C04
Table 1051: EP_OUT_DONEQ_WRITE_POINTER_REG_4 Register 963
Offset: 0x00050C08
Table 1052: EP_OUT_DONEQ_WRITE_POINTER_REG_5 Register 963
Offset: 0x00050C0C
Table 1053: EP_OUT_DONEQ_WRITE_POINTER_REG_6 Register 963
Offset: 0x00050C10
Table 1054: EP_OUT_DONEQ_WRITE_POINTER_REG_7 Register 963
Offset: 0x00050C14
Table 1055: EP_OUT_DONEQ_WRITE_POINTER_REG_8 Register 964
Offset: 0x00050C18
Table 1056: EP_OUT_DONEQ_WRITE_POINTER_REG_9 Register 964
Offset: 0x00050C1C
Table 1057: EP_OUT_DONEQ_WRITE_POINTER_REG_10 Register 964
Offset: 0x00050C20
Table 1058: EP_OUT_DONEQ_WRITE_POINTER_REG_11 Register 964
Offset: 0x00050C24
Table 1059: EP_OUT_DONEQ_WRITE_POINTER_REG_12 Register 964
Offset: 0x00050C28
Table 1060: EP_OUT_DONEQ_WRITE_POINTER_REG_13 Register 965
Offset: 0x00050C2C
Table 1061: EP_OUT_DONEQ_WRITE_POINTER_REG_14 Register 965
Offset: 0x00050C30
Table 1062: EP_OUT_DONEQ_WRITE_POINTER_REG_15 Register 965
Offset: 0x00050C34
Table 1063: EP_OUT_DONEQ_READ_POINTER_REG_0 Register 965
Offset: 0x00050C38
Table 1064: EP_OUT_DONEQ_READ_POINTER_REG_1 Register 965
Offset: 0x00050C3C
Table 1065: EP_OUT_DONEQ_READ_POINTER_REG_2 Register 966
Offset: 0x00050C40
Table 1066: EP_OUT_DONEQ_READ_POINTER_REG_3 Register 966
Offset: 0x00050C44

Copyright © 2015 Marvell

Table 1067: EP_OUT_DONEQ_READ_POINTER_REG_4 Register 966
Offset: 0x00050C48
Table 1068: EP_OUT_DONEQ_READ_POINTER_REG_5 Register 966
Offset: 0x00050C4C
Table 1069: EP_OUT_DONEQ_READ_POINTER_REG_6 Register 967
Offset: 0x00050C50
Table 1070: EP_OUT_DONEQ_READ_POINTER_REG_7 Register 967
Offset: 0x00050C54
Table 1071: EP_OUT_DONEQ_READ_POINTER_REG_8 Register 967
Offset: 0x00050C58
Table 1072: EP_OUT_DONEQ_READ_POINTER_REG_9 Register 967
Offset: 0x00050C5C
Table 1073: EP_OUT_DONEQ_READ_POINTER_REG_10 Register 968
Offset: 0x00050C60
Table 1074: EP_OUT_DONEQ_READ_POINTER_REG_11 Register 968
Offset: 0x00050C64
Table 1075: EP_OUT_DONEQ_READ_POINTER_REG_12 Register 968
Offset: 0x00050C68
Table 1076: EP_OUT_DONEQ_READ_POINTER_REG_13 Register 968
Offset: 0x00050C6C
Table 1077: EP_OUT_DONEQ_READ_POINTER_REG_14 Register 969
Offset: 0x00050C70
Table 1078: EP_OUT_DONEQ_READ_POINTER_REG_15 Register 969
Offset: 0x00050C74
Table 1079: DONEQ_FULL_STATUS_REG Register 969
Offset: 0x00050C78
Table 1080: ABORT_REQ_STATUS_REG Register 974
Offset: 0x00050C7C
Table 1081: ABORT_ID_REG_1 Register 977
Offset: 0x00050C84
Table 1082: ABORT_ID_REG_2 Register 978
Offset: 0x00050C88
Table 1083: ABORT_ID_REG_3 Register 978
Offset: 0x00050C8C
Table 1084: ABORT_ID_REG_4 Register 978
Offset: 0x00050C90
Table 1085: ABORT_ID_REG_5 Register 979
Offset: 0x00050C94
Table 1086: ABORT_ID_REG_6 Register 979
Offset: 0x00050C98
Table 1087: ABORT_ID_REG_7 Register 980
Offset: 0x00050C9C
Table 1088: ABORT_ID_REG_8 Register 980
Offset: 0x00050CA0
Table 1089: ABORT_ID_REG_9 Register 980
Offset: 0x00050CA4
Table 1090: ABORT_ID_REG_10 Register 981
Offset: 0x00050CA8

Document Classification: Proprietary Information

Table 1091: ABORT_ID_REG_11 Register 981
Offset: 0x00050CAC
Table 1092: ABORT_ID_REG_12 Register 982
Offset: 0x00050CB0
Table 1093: ABORT_ID_REG_13 Register 982
Offset: 0x00050CB4
Table 1094: ABORT_ID_REG_14 Register 982
Offset: 0x00050CB8
Table 1095: ABORT_ID_REG_15 Register 983
Offset: 0x00050CBC
Table 1096: FLUSHED_BD_COUNTER_REG_1 Register 983
Offset: 0x00050CC0
Table 1097: FLUSHED_BD_COUNTER_REG_2 Register 984
Offset: 0x00050CC4
Table 1098: FLUSHED_BD_COUNTER_REG_3 Register 984
Offset: 0x00050CC8
Table 1099: FLUSHED_BD_COUNTER_REG_4 Register 984
Offset: 0x00050CCC
Table 1100: FLUSHED_BD_COUNTER_REG_5 Register 985
Offset: 0x00050CD0
Table 1101: FLUSHED_BD_COUNTER_REG_6 Register 985
Offset: 0x00050CD4
Table 1102: FLUSHED_BD_COUNTER_REG_7 Register 985
Offset: 0x00050CD8
Table 1103: FLUSHED_BD_COUNTER_REG_8 Register 986
Offset: 0x00050CDC
Table 1104: FLUSHED_BD_COUNTER_REG_9 Register 986
Offset: 0x00050CE0
Table 1105: FLUSHED_BD_COUNTER_REG_10 Register 986
Offset: 0x00050CE4
Table 1106: FLUSHED_BD_COUNTER_REG_11 Register 987
Offset: 0x00050CE8
Table 1107: FLUSHED_BD_COUNTER_REG_12 Register 987
Offset: 0x00050CEC
Table 1108: FLUSHED_BD_COUNTER_REG_13 Register 987
Offset: 0x00050CF0
Table 1109: FLUSHED_BD_COUNTER_REG_14 Register 988
Offset: 0x00050CF4
Table 1110: FLUSHED_BD_COUNTER_REG_15 Register 988
Offset: 0x00050CF8
Table 1111: EP_IN_FLUSHED_SID_BITMAP_REG_1 Register 988
Offset: 0x00050CFC
Table 1112: EP_IN_FLUSHED_SID_BITMAP_REG_2 Register 988
Offset: 0x00050D1C
Table 1113: EP_IN_FLUSHED_SID_BITMAP_REG_3 Register 989
Offset: 0x00050D3C
Table 1114: EP_IN_FLUSHED_SID_BITMAP_REG_4 Register 989
Offset: 0x00050D5C

Copyright © 2015 Marvell

Table 1115: EP_IN_FLUSHED_SID_BITMAP_REG_5 Register 989
Offset: 0x00050D7C
Table 1116: EP_IN_FLUSHED_SID_BITMAP_REG_6 Register 989
Offset: 0x00050D9C
Table 1117: EP_IN_FLUSHED_SID_BITMAP_REG_7 Register 989
Offset: 0x00050DBC
Table 1118: EP_IN_FLUSHED_SID_BITMAP_REG_8 Register 990
Offset: 0x00050DDC
Table 1119: EP_IN_FLUSHED_SID_BITMAP_REG_9 Register 990
Offset: 0x00050DFC
Table 1120: EP_IN_FLUSHED_SID_BITMAP_REG_10 Register 990
Offset: 0x00050E1C
Table 1121: EP_IN_FLUSHED_SID_BITMAP_REG_11 Register 990
Offset: 0x00050E3C
Table 1122: EP_IN_FLUSHED_SID_BITMAP_REG_12 Register 990
Offset: 0x00050E5C
Table 1123: EP_IN_FLUSHED_SID_BITMAP_REG_13 Register 991
Offset: 0x00050E7C
Table 1124: EP_IN_FLUSHED_SID_BITMAP_REG_14 Register 991
Offset: 0x00050E9C
Table 1125: EP_IN_FLUSHED_SID_BITMAP_REG_15 Register 991
Offset: 0x00050EBC
Table 1126: EP_OUT_FLUSHED_SID_BITMAP_REG_1 Register 991
Offset: 0x00050EDC
Table 1127: EP_OUT_FLUSHED_SID_BITMAP_REG_2 Register 991
Offset: 0x00050EFC
Table 1128: EP_OUT_FLUSHED_SID_BITMAP_REG_3 Register 992
Offset: 0x00050F1C
Table 1129: EP_OUT_FLUSHED_SID_BITMAP_REG_4 Register 992
Offset: 0x00050F3C
Table 1130: EP_OUT_FLUSHED_SID_BITMAP_REG_5 Register 992
Offset: 0x00050F5C
Table 1131: EP_OUT_FLUSHED_SID_BITMAP_REG_6 Register 992
Offset: 0x00050F7C
Table 1132: EP_OUT_FLUSHED_SID_BITMAP_REG_7 Register 992
Offset: 0x00050F9C
Table 1133: EP_OUT_FLUSHED_SID_BITMAP_REG_8 Register 993
Offset: 0x00050FBC
Table 1134: EP_OUT_FLUSHED_SID_BITMAP_REG_9 Register 993
Offset: 0x00050FDC
Table 1135: EP_OUT_FLUSHED_SID_BITMAP_REG_10 Register 993
Offset: 0x00050FFC
Table 1136: EP_OUT_FLUSHED_SID_BITMAP_REG_11 Register 993
Offset: 0x0005101C
Table 1137: EP_OUT_FLUSHED_SID_BITMAP_REG_12 Register 993
Offset: 0x0005103C
Table 1138: EP_OUT_FLUSHED_SID_BITMAP_REG_13 Register 994
Offset: 0x0005105C

Copyright © 2015 Marvell

Table 1139: EP_OUT_FLUSHED_SID_BITMAP_REG_14 Register 994
Offset: 0x0005107C
Table 1140: EP_OUT_FLUSHED_SID_BITMAP_REG_15 Register 994
Offset: 0x0005109C
Table 1141: EP_IN_NRDYQ_HEAD_POINTER_REG_1 Register 994
Offset: 0x000510BC
Table 1142: EP_IN_NRDYQ_HEAD_POINTER_REG_2 Register 994
Offset: 0x000510C0
Table 1143: EP_IN_NRDYQ_HEAD_POINTER_REG_3 Register 995
Offset: 0x000510C4
Table 1144: EP_IN_NRDYQ_HEAD_POINTER_REG_4 Register 995
Offset: 0x000510C8
Table 1145: EP_IN_NRDYQ_HEAD_POINTER_REG_5 Register 995
Offset: 0x000510CC
Table 1146: EP_IN_NRDYQ_HEAD_POINTER_REG_6 Register 995
Offset: 0x000510D0
Table 1147: EP_IN_NRDYQ_HEAD_POINTER_REG_7 Register 995
Offset: 0x000510D4
Table 1148: EP_IN_NRDYQ_HEAD_POINTER_REG_8 Register 996
Offset: 0x000510D8
Table 1149: EP_IN_NRDYQ_HEAD_POINTER_REG_9 Register 996
Offset: 0x000510DC
Table 1150: EP_IN_NRDYQ_HEAD_POINTER_REG_10 Register 996
Offset: 0x000510E0
Table 1151: EP_IN_NRDYQ_HEAD_POINTER_REG_11 Register 996
Offset: 0x000510E4
Table 1152: EP_IN_NRDYQ_HEAD_POINTER_REG_12 Register 996
Offset: 0x000510E8
Table 1153: EP_IN_NRDYQ_HEAD_POINTER_REG_13 Register 997
Offset: 0x000510EC
Table 1154: EP_IN_NRDYQ_HEAD_POINTER_REG_14 Register 997
Offset: 0x000510F0
Table 1155: EP_IN_NRDYQ_HEAD_POINTER_REG_15 Register 997
Offset: 0x000510F4
Table 1156: EP_OUT_NRDYQ_HEAD_POINTER_REG_1 Register 997
Offset: 0x000510F8
Table 1157: EP_OUT_NRDYQ_HEAD_POINTER_REG_2 Register 997
Offset: 0x000510FC
Table 1158: EP_OUT_NRDYQ_HEAD_POINTER_REG_3 Register 998
Offset: 0x00051100
Table 1159: EP_OUT_NRDYQ_HEAD_POINTER_REG_4 Register 998
Offset: 0x00051104
Table 1160: EP_OUT_NRDYQ_HEAD_POINTER_REG_5 Register 998
Offset: 0x00051108
Table 1161: EP_OUT_NRDYQ_HEAD_POINTER_REG_6 Register 998
Offset: 0x0005110C
Table 1162: EP_OUT_NRDYQ_HEAD_POINTER_REG_7 Register 998
Offset: 0x00051110

Copyright © 2015 Marvell

Table 1163: EP_OUT_NRDYQ_HEAD_POINTER_REG_8 Register 999
Offset: 0x00051114
Table 1164: EP_OUT_NRDYQ_HEAD_POINTER_REG_9 Register 999
Offset: 0x00051118
Table 1165: EP_OUT_NRDYQ_HEAD_POINTER_REG_10 Register 999
Offset: 0x0005111C
Table 1166: EP_OUT_NRDYQ_HEAD_POINTER_REG_11 Register 999
Offset: 0x00051120
Table 1167: EP_OUT_NRDYQ_HEAD_POINTER_REG_12 Register 999
Offset: 0x00051124
Table 1168: EP_OUT_NRDYQ_HEAD_POINTER_REG_13 Register 1000
Offset: 0x00051128
Table 1169: EP_OUT_NRDYQ_HEAD_POINTER_REG_14 Register 1000
Offset: 0x0005112C
Table 1170: EP_OUT_NRDYQ_HEAD_POINTER_REG_15 Register 1000
Offset: 0x00051130
Table 1171: EP_IN_NRDYQ_TAIL_POINTER_REG_1 Register 1000
Offset: 0x00051134
Table 1172: EP_IN_NRDYQ_TAIL_POINTER_REG_2 Register 1000
Offset: 0x00051138
Table 1173: EP_IN_NRDYQ_TAIL_POINTER_REG_3 Register 1001
Offset: 0x0005113C
Table 1174: EP_IN_NRDYQ_TAIL_POINTER_REG_4 Register 1001
Offset: 0x00051140
Table 1175: EP_IN_NRDYQ_TAIL_POINTER_REG_5 Register 1001
Offset: 0x00051144
Table 1176: EP_IN_NRDYQ_TAIL_POINTER_REG_6 Register 1001
Offset: 0x00051148
Table 1177: EP_IN_NRDYQ_TAIL_POINTER_REG_7 Register 1001
Offset: 0x0005114C
Table 1178: EP_IN_NRDYQ_TAIL_POINTER_REG_8 Register 1002
Offset: 0x00051150
Table 1179: EP_IN_NRDYQ_TAIL_POINTER_REG_9 Register 1002
Offset: 0x00051154
Table 1180: EP_IN_NRDYQ_TAIL_POINTER_REG_10 Register 1002
Offset: 0x00051158
Table 1181: EP_IN_NRDYQ_TAIL_POINTER_REG_11 Register 1002
Offset: 0x0005115C
Table 1182: EP_IN_NRDYQ_TAIL_POINTER_REG_12 Register 1002
Offset: 0x00051160
Table 1183: EP_IN_NRDYQ_TAIL_POINTER_REG_13 Register 1003
Offset: 0x00051164
Table 1184: EP_IN_NRDYQ_TAIL_POINTER_REG_14 Register 1003
Offset: 0x00051168
Table 1185: EP_IN_NRDYQ_TAIL_POINTER_REG_15 Register 1003
Offset: 0x0005116C
Table 1186: EP_OUT_NRDYQ_TAIL_POINTER_REG_1 Register 1003
Offset: 0x00051170

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1187: EP_OUT_NRDYQ_TAIL_POINTER_REG_2 Register 1003
Offset: 0x00051174
Table 1188: EP_OUT_NRDYQ_TAIL_POINTER_REG_3 Register 1004
Offset: 0x00051178
Table 1189: EP_OUT_NRDYQ_TAIL_POINTER_REG_4 Register 1004
Offset: 0x0005117C
Table 1190: EP_OUT_NRDYQ_TAIL_POINTER_REG_5 Register 1004
Offset: 0x00051180
Table 1191: EP_OUT_NRDYQ_TAIL_POINTER_REG_6 Register 1004
Offset: 0x00051184
Table 1192: EP_OUT_NRDYQ_TAIL_POINTER_REG_7 Register 1004
Offset: 0x00051188
Table 1193: EP_OUT_NRDYQ_TAIL_POINTER_REG_8 Register 1005
Offset: 0x0005118C
Table 1194: EP_OUT_NRDYQ_TAIL_POINTER_REG_9 Register 1005
Offset: 0x00051190
Table 1195: EP_OUT_NRDYQ_TAIL_POINTER_REG_10 Register 1005
Offset: 0x00051194
Table 1196: EP_OUT_NRDYQ_TAIL_POINTER_REG_11 Register 1005
Offset: 0x00051198
Table 1197: EP_OUT_NRDYQ_TAIL_POINTER_REG_12 Register 1005
Offset: 0x0005119C
Table 1198: EP_OUT_NRDYQ_TAIL_POINTER_REG_13 Register 1006
Offset: 0x000511A0
Table 1199: EP_OUT_NRDYQ_TAIL_POINTER_REG_14 Register 1006
Offset: 0x000511A4
Table 1200: EP_OUT_NRDYQ_TAIL_POINTER_REG_15 Register 1006
Offset: 0x000511A8
Table 1201: EP_IN_RDYQ_HEAD_POINTER_REG_1 Register 1006
Offset: 0x000511AC
Table 1202: EP_IN_RDYQ_HEAD_POINTER_REG_2 Register 1006
Offset: 0x000511B0
Table 1203: EP_IN_RDYQ_HEAD_POINTER_REG_3 Register 1007
Offset: 0x000511B4
Table 1204: EP_IN_RDYQ_HEAD_POINTER_REG_4 Register 1007
Offset: 0x000511B8
Table 1205: EP_IN_RDYQ_HEAD_POINTER_REG_5 Register 1007
Offset: 0x000511BC
Table 1206: EP_IN_RDYQ_HEAD_POINTER_REG_6 Register 1007
Offset: 0x000511C0
Table 1207: EP_IN_RDYQ_HEAD_POINTER_REG_7 Register 1007
Offset: 0x000511C4
Table 1208: EP_IN_RDYQ_HEAD_POINTER_REG_8 Register 1008
Offset: 0x000511C8
Table 1209: EP_IN_RDYQ_HEAD_POINTER_REG_9 Register 1008
Offset: 0x000511CC
Table 1210: EP_IN_RDYQ_HEAD_POINTER_REG_10 Register 1008
Offset: 0x000511D0

Copyright © 2015 Marvell

Table 1211: EP_IN_RDYQ_HEAD_POINTER_REG_11 Register 1008
Offset: 0x000511D4
Table 1212: EP_IN_RDYQ_HEAD_POINTER_REG_12 Register 1008
Offset: 0x000511D8
Table 1213: EP_IN_RDYQ_HEAD_POINTER_REG_13 Register 1009
Offset: 0x000511DC
Table 1214: EP_IN_RDYQ_HEAD_POINTER_REG_14 Register 1009
Offset: 0x000511E0
Table 1215: EP_IN_RDYQ_HEAD_POINTER_REG_15 Register 1009
Offset: 0x000511E4
Table 1216: EP_OUT_RDYQ_HEAD_POINTER_REG_1 Register 1009
Offset: 0x000511E8
Table 1217: EP_OUT_RDYQ_HEAD_POINTER_REG_2 Register 1009
Offset: 0x000511EC
Table 1218: EP_OUT_RDYQ_HEAD_POINTER_REG_3 Register 1010
Offset: 0x000511F0
Table 1219: EP_OUT_RDYQ_HEAD_POINTER_REG_4 Register 1010
Offset: 0x000511F4
Table 1220: EP_OUT_RDYQ_HEAD_POINTER_REG_5 Register 1010
Offset: 0x000511F8
Table 1221: EP_OUT_RDYQ_HEAD_POINTER_REG_6 Register 1010
Offset: 0x000511FC
Table 1222: EP_OUT_RDYQ_HEAD_POINTER_REG_7 Register 1010
Offset: 0x00051200
Table 1223: EP_OUT_RDYQ_HEAD_POINTER_REG_8 Register 1011
Offset: 0x00051204
Table 1224: EP_OUT_RDYQ_HEAD_POINTER_REG_9 Register 1011
Offset: 0x00051208
Table 1225: EP_OUT_RDYQ_HEAD_POINTER_REG_10 Register 1011
Offset: 0x0005120C
Table 1226: EP_OUT_RDYQ_HEAD_POINTER_REG_11 Register 1011
Offset: 0x00051210
Table 1227: EP_OUT_RDYQ_HEAD_POINTER_REG_12 Register 1011
Offset: 0x00051214
Table 1228: EP_OUT_RDYQ_HEAD_POINTER_REG_13 Register 1012
Offset: 0x00051218
Table 1229: EP_OUT_RDYQ_HEAD_POINTER_REG_14 Register 1012
Offset: 0x0005121C
Table 1230: EP_OUT_RDYQ_HEAD_POINTER_REG_15 Register 1012
Offset: 0x00051220
Table 1231: BULK_STREAMING_INT_STATUS_REG Register 1012
Offset: 0x00051224
Table 1232: BULK_STREAMING_INT_ENABLE_REG Register 1013
Offset: 0x00051228
Table 1233: PRIME_RCV_STATUS_REG Register 1013
Offset: 0x0005122C
Table 1234: HIMD_RCV_STATUS_REG Register 1015
Offset: 0x00051230

Copyright © 2015 Marvell

Table 1235: STREAM_REJ_STATUS_REG Register 1016
Offset: 0x00051234
Table 1236: HOST_FLOWCTRL_STATUS_REG Register 1018
Offset: 0x00051238
Table 1237: BD_MEM_IN_REG_1 Register 1020
Offset: 0x0005123C
Table 1238: PL_DEBUG_REG_1 Register 1020
Offset: 0x00051264
Table 1239: PL_DEBUG_REG_2 Register 1020
Offset: 0x00051268
Table 1240: PL_DEBUG_REG_3 Register 1021
Offset: 0x0005126C
Table 1241: PL_DEBUG_REG_4 Register 1021
Offset: 0x00051270
Table 1242: DMA_DBG_IN_REG_1 Register 1021
Offset: 0x00051274
Table 1243: DMA_DBG_IN_REG_2 Register 1021
Offset: 0x00051278
Table 1244: DMA_DBG_OUT_REG_1 Register 1022
Offset: 0x0005127C
Table 1245: DMA_DBG_OUT_REG_2 Register 1022
Offset: 0x00051280
Table 1246: USB2_USBSTS Register 1022
Offset: 0x00054144
Table 1247: USB2_USBINT Register 1023
Offset: 0x00054148
Table 1248: USB2_DEVLC Register 1024
Offset: 0x0005415C
Table 1249: USB2_PORTSC Register 1025
Offset: 0x00054184

A.14 SATA 3.0 Host Controller Registers 1027

Table 1251: SoC Capability Register (i=0–1) 1029
Offset: Unit0: 0x000A8000, Unit1: 0x000E0000
Table 1252: Global SoC Control Register (i=0–1) 1030
Offset: Unit0: 0x000A8004, Unit1: 0x000E0004
Table 1253: Interrupt Status Register (i=0–1) 1030
Offset: Unit0: 0x000A8008, Unit1: 0x000E0008
Table 1254: Ports Implemented Register (i=0–1) 1031
Offset: Unit0: 0x000A800C, Unit1: 0x000E000C
Table 1255: AHCI Version Register (i=0–1) 1031
Offset: Unit0: 0x000A8010, Unit1: 0x000E0010
Table 1256: Command Completion Coalescing Control Register (i=0–1) 1031
Offset: Unit0: 0x000A8014, Unit1: 0x000E0014
Table 1257: Command Completion Coalescing Ports Register (i=0–1) 1032
Offset: Unit0: 0x000A8018, Unit1: 0x000E0018
Table 1258: Vendor-Specific 0 Address Register (i=0–1) 1033
Offset: Unit0: 0x000A80A0, Unit1: 0x000E00A0

Copyright © 2015 Marvell

Table 1259: Vendor-Specific 0 Data Register (i=0–1) 1033
Offset: Unit0: 0x000A80A4, Unit1: 0x000E00A4
Table 1260: Window0 Control Register (i=0–1) 1033
Offset: Unit0: 0x000A8060, Unit1: 0x000E0060
Table 1261: Window0 Base Register (i=0–1) 1034
Offset: Unit0: 0x000A8064, Unit1: 0x000E0064
Table 1262: Window0 Size Register (i=0–1) 1034
Offset: Unit0: 0x000A8068, Unit1: 0x000E0068
Table 1263: Window1 Control Register (i=0–1) 1035
Offset: Unit0: 0x000A8070, Unit1: 0x000E0070
Table 1264: Window1 Base Register (i=0–1) 1035
Offset: Unit0: 0x000A8074, Unit1: 0x000E0074
Table 1265: Window1 Size Register (i=0–1) 1035
Offset: Unit0: 0x000A8078, Unit1: 0x000E0078
Table 1266: Window2 Control Register (i=0–1) 1035
Offset: Unit0: 0x000A8080, Unit1: 0x000E0080
Table 1267: Window2 Base Register (i=0–1) 1036
Offset: Unit0: 0x000A8084, Unit1: 0x000E0084
Table 1268: Window2 Size Register (i=0–1) 1036
Offset: Unit0: 0x000A8088, Unit1: 0x000E0088
Table 1269: Window3 Control Register (i=0–1) 1036
Offset: Unit0: 0x000A8090, Unit1: 0x000E0090
Table 1270: Window3 Base Register (i=0–1) 1037
Offset: Unit0: 0x000A8094, Unit1: 0x000E0094
Table 1271: Window3 Size Register (i=0–1) 1037
Offset: Unit0: 0x000A8098, Unit1: 0x000E0098
Table 1272: Command List Base Address Lower 32-bits Register (i=0–1, n=0–1) 1037
Offset: Unit0 Port0: 0x000A8100, Unit0 Port1: 0x000A8180, Unit1 Port0: 0x000E0100, Unit1 Port1: 0x000E0180
Table 1273: SATA control register (i=0–1, n=0–1) 1038
Offset: Unit0 Port0: 0x000A8100, Unit0 Port1: 0x000A8180, Unit1 Port0: 0x000E0100, Unit1 Port1: 0x000E0180
Table 1274: MBUS request size and interface select register (i=0–1, n=0–1) 1039
Offset: Unit0 Port0: 0x000A8104, Unit0 Port1: 0x000A8184, Unit1 Port0: 0x000E0104, Unit1 Port1: 0x000E0184
Table 1275: Command List Base Address Upper 32-bits Register (i=0–1, n=0–1) 1039
Offset: Unit0 Port0: 0x000A8104, Unit0 Port1: 0x000A8184, Unit1 Port0: 0x000E0104, Unit1 Port1: 0x000E0184
Table 1276: SATA memory timing control register (i=0–1, n=0–1) 1039
Offset: Unit0 Port0: 0x000A8108, Unit0 Port1: 0x000A8188, Unit1 Port0: 0x000E0108, Unit1 Port1: 0x000E0188
Table 1277: Received FIS Base Address Lower 32-bits Register (i=0–1, n=0–1) 1040
Offset: Unit0 Port0: 0x000A8108, Unit0 Port1: 0x000A8188, Unit1 Port0: 0x000E0108, Unit1 Port1: 0x000E0188
Table 1278: Received FIS Base Address Upper 32-bits Register (i=0–1, n=0–1) 1040
Offset: Unit0 Port0: 0x000A810C, Unit0 Port1: 0x000A818C, Unit1 Port0: 0x000E010C, Unit1 Port1: 0x000E018C
Table 1279: Port Interrupt Status Register (i=0–1, n=0–1) 1040
Offset: Unit0 Port0: 0x000A8110, Unit0 Port1: 0x000A8190, Unit1 Port0: 0x000E0110, Unit1 Port1: 0x000E0190

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1280: Port Interrupt Enable Register (i=0–1, n=0–1) 1043
Offset: Unit0 Port0: 0x000A8114, Unit0 Port1: 0x000A8194, Unit1 Port0: 0x000E0114, Unit1 Port1: 0x000E0194
Table 1281: Port Command and Status Register (i=0–1, n=0–1) 1045
Offset: Unit0 Port0: 0x000A8118, Unit0 Port1: 0x000A8198, Unit1 Port0: 0x000E0118, Unit1 Port1: 0x000E0198
Table 1282: Reserved 1 Register (i=0–1, n=0–1) 1047
Offset: Unit0 Port0: 0x000A811C, Unit0 Port1: 0x000A819C, Unit1 Port0: 0x000E011C, Unit1 Port1: 0x000E019C
Table 1283: Port Task File Data Register (i=0–1, n=0–1) 1047
Offset: Unit0 Port0: 0x000A8120, Unit0 Port1: 0x000A81A0, Unit1 Port0: 0x000E0120, Unit1 Port1: 0x000E01A0
Table 1284: Port Signature Register (i=0–1, n=0–1) 1048
Offset: Unit0 Port0: 0x000A8124, Unit0 Port1: 0x000A81A4, Unit1 Port0: 0x000E0124, Unit1 Port1: 0x000E01A4
Table 1285: SATA Status (SStatus/SCR0) Register (i=0–1, n=0–1) 1048
Offset: Unit0 Port0: 0x000A8128, Unit0 Port1: 0x000A81A8, Unit1 Port0: 0x000E0128, Unit1 Port1: 0x000E01A8
Table 1286: SATA Control (SControl/SCR2) Register (i=0–1, n=0–1) 1049
Offset: Unit0 Port0: 0x000A812C, Unit0 Port1: 0x000A81AC, Unit1 Port0: 0x000E012C, Unit1 Port1: 0x000E01AC
Table 1287: SATA Error (SError/SCR1) Register (i=0–1, n=0–1) 1050
Offset: Unit0 Port0: 0x000A8130, Unit0 Port1: 0x000A81B0, Unit1 Port0: 0x000E0130, Unit1 Port1: 0x000E01B0
Table 1288: SATA Active (SActive/SCR3) Register (i=0–1, n=0–1) 1052
Offset: Unit0 Port0: 0x000A8134, Unit0 Port1: 0x000A81B4, Unit1 Port0: 0x000E0134, Unit1 Port1: 0x000E01B4
Table 1289: Command Issue Register (i=0–1, n=0–1) 1052
Offset: Unit0 Port0: 0x000A8138, Unit0 Port1: 0x000A81B8, Unit1 Port0: 0x000E0138, Unit1 Port1: 0x000E01B8
Table 1290: Port SATA Notification (SNotification/SCR4) Register (i=0–1, n=0–1) 1052
Offset: Unit0 Port0: 0x000A813C, Unit0 Port1: 0x000A81BC, Unit1 Port0: 0x000E013C, Unit1 Port1: 0x000E01BC
Table 1291: Port FIS-Based Switching Control Register (i=0–1, n=0–1) 1053
Offset: Unit0 Port0: 0x000A8140, Unit0 Port1: 0x000A81C0, Unit1 Port0: 0x000E0140, Unit1 Port1: 0x000E01C0

A.15 Audio (I2S / S/PDIF) Registers 1054

Table 1293: Recording Window Base Address Register 1055
Offset: 0x000E8A00
Table 1294: Recording Window Control Register 1055
Offset: 0x000E8A04
Table 1295: Playback Window Base Address Register 1056
Offset: 0x000E8A08
Table 1296: Playback Window Control Register 1056
Offset: 0x000E8A0C
Table 1297: Sample Counters Control Register 1057
Offset: 0x000E9220
Table 1298: Playback Sample Counter Register 1057
Offset: 0x000E9224

Copyright © 2015 Marvell

Table 1299: Recording Sample Counter Register 1058
Offset: 0x000E9228
Table 1300: Clocks Control Register 1058
Offset: 0x000E9230
Table 1301: Reserved Register 1058
Offset: 0x000E9238
Table 1302: Audio Error Cause Register 1059
Offset: 0x000E9300
Table 1303: Audio Error Mask Register 1059
Offset: 0x000E9304
Table 1304: Audio Interrupt Cause Register 1059
Offset: 0x000E9308
Table 1305: Audio Interrupt Mask Register 1061
Offset: 0x000E930C
Table 1306: Recorded Byte Count for Interrupt Register 1061
Offset: 0x000E9310
Table 1307: Playback Byte Count for Interrupt Register 1061
Offset: 0x000E9314
Table 1308: Playback Byte Counter for Interrupt Register 1061
Offset: 0x000E9318
Table 1309: Recorded Byte Counter for Interrupt Register 1062
Offset: 0x000E931C
Table 1310: Playback Control Register 1062
Offset: 0x000E9100
Table 1311: Playback Start Address Register 1065
Offset: 0x000E9104
Table 1312: Playback Buffer Size Register 1065
Offset: 0x000E9108
Table 1313: Playback Buffer Byte Counter Register 1065
Offset: 0x000E910C
Table 1314: SPDIF Playback Control Register 1066
Offset: 0x000EA204
Table 1315: SPDIF Playback Channel Status Left n Register (n=0–5) 1067
Offset: status0: 0x000EA280, status1: 0x000EA284, status2: 0x000EA288, status3: 0x000EA28C, status4: 0x000EA290, status5: 0x000EA294
Table 1316: SPDIF Playback Channel Status Right n Register (n=0–5) 1067
Offset: status0: 0x000EA2A0, status1: 0x000EA2A4, status2: 0x000EA2A8, status3: 0x000EA2AC, status4: 0x000EA2B0, status5: 0x000EA2B4
Table 1317: SPDIF Playback User Bits Left n Register (n=0–5) 1067
Offset: User0: 0x000EA2C0, User1: 0x000EA2C4, User2: 0x000EA2C8, User3: 0x000EA2CC, User4: 0x000EA2D0, User5: 0x000EA2D4
Table 1318: SPDIF Playback User Bits Right n Register (n=0–5) 1067
Offset: User0: 0x000EA2E0, User1: 0x000EA2E4, User2: 0x000EA2E8, User3: 0x000EA2EC, User4: 0x000EA2F0, User5: 0x000EA2F4
Table 1319: I2S Playback Control Register 1068
Offset: 0x000EA508
Table 1320: Recording Control Register 1068
Offset: 0x000E9000

Copyright © 2015 Marvell

Table 1321: Recording Start Address Register 1070
Offset: 0x000E9004
Table 1322: Recording Buffer Size Register 1070
Offset: 0x000E9008
Table 1323: Recording Buffer Byte Counter Register 1071
Offset: 0x000E900C
Table 1324: I2S Recording Control Register 1071
Offset: 0x000EA408

A.16 Serial Peripheral Interface (SPI) Registers 1073

Table 1326: SPI Control Register (N=0–1) 1074
Offset: SPI_Controller0: 0x00010600, SPI_Controller1: 0x00010680
Table 1327: SPI Interface Configuration Register (N=0–1) 1074
Offset: SPI_Controller0: 0x00010604, SPI_Controller1: 0x00010684
Table 1328: SPI Data Out Register (N=0–1) 1076
Offset: SPI_Controller0: 0x00010608, SPI_Controller1: 0x00010688
Table 1329: SPI Data In Register (N=0–1) 1076
Offset: SPI_Controller0: 0x0001060C, SPI_Controller1: 0x0001068C
Table 1330: SPI Interrupt Cause Register (N=0–1) 1076
Offset: SPI_Controller0: 0x00010610, SPI_Controller1: 0x00010690
Table 1331: SPI Interrupt Mask Register (N=0–1) 1076
Offset: SPI_Controller0: 0x00010614, SPI_Controller1: 0x00010694
Table 1332: SPI Timing Parameters 1 Register (N=0–1) 1077
Offset: SPI_Controller0: 0x00010618, SPI_Controller1: 0x00010698
Table 1333: SPI Timing Parameters 2 Register (N=0–1) 1078
Offset: SPI_Controller0: 0x0001061C, SPI_Controller1: 0x0001069C
Table 1334: SPI Direct Write Configuration Register (N=0–1) 1078
Offset: SPI_Controller0: 0x00010620, SPI_Controller1: 0x000106A0
Table 1335: SPI Direct Write Header Register (N=0–1) 1079
Offset: SPI_Controller0: 0x00010624, SPI_Controller1: 0x000106A4
Table 1336: SPI Direct Read Header Register (N=0–1) 1079
Offset: SPI_Controller0: 0x00010628, SPI_Controller1: 0x000106A8
Table 1337: SPI CS Address Decode Register (N=0–1) 1079
Offset: SPI_Controller0: 0x0001062C, SPI_Controller1: 0x000106AC
Table 1338: SPI CSn Timing Parameters Register (n=0–7, N=0–1) 1080
Offset: SPI_Controller0SPI CS0: 0x00010630, SPI_Controller0SPI CS1: 0x00010634...SPI_Controller1SPI CS7: 0x000106CC
Table 1339: SPI Controller Version Register (N=0–1) 1081
Offset: SPI_Controller0: 0x00010650, SPI_Controller1: 0x000106D0

A.17 Secure Digital Input/Output (SDIO) 3.0 Registers 1082

Table 1341: System Address Register 1083
Offset: 0x000D8000
Table 1342: Block Count and Size Register 1084
Offset: 0x000D8004
Table 1343: Command Argument Register 1084
Offset: 0x000D8008

Table 1344: Command and Transfer Mode Register 1085
Offset: 0x000D800C
Table 1345: Response 0 and 1 Register 1086
Offset: 0x000D8010
Table 1346: Response 2 and 3 Register 1087
Offset: 0x000D8014
Table 1347: Response 4 and 5 Register 1087
Offset: 0x000D8018
Table 1348: Response 6 and 7 Register 1087
Offset: 0x000D801C
Table 1349: Buffer Data Port 0 and 1 Register 1088
Offset: 0x000D8020
Table 1350: Present State Register 1088
Offset: 0x000D8024
Table 1351: Block Gap and Host Control Register 1090
Offset: 0x000D8028
Table 1352: Timeout Control/Software Reset and Clock Control Register 1093
Offset: 0x000D802C
Table 1353: Normal and Error Interrupt Status Register 1095
Offset: 0x000D8030
Table 1354: Normal and Error Interrupt Status Enable Register 1097
Offset: 0x000D8034
Table 1355: Normal and Error Interrupt Status Interrupt Enable Register 1099
Offset: 0x000D8038
Table 1356: Auto Host Control2 and Command12 Error Status Register 1102
Offset: 0x000D803C
Table 1357: Capabilities 1 and 0 Register 1103
Offset: 0x000D8040
Table 1358: Capabilities 2 and 3 Register 1105
Offset: 0x000D8044
Table 1359: Maximum Current 1 and 2 Register 1106
Offset: 0x000D8048
Table 1360: Maximum Current 3 and 4 Register 1107
Offset: 0x000D804C
Table 1361: Force Event Auto Command 12 Error Register 1107
Offset: 0x000D8050
Table 1362: ADMA Error Status Register 1108
Offset: 0x000D8054
Table 1363: ADMA System Address Register High 1109
Offset: 0x000D8058
Table 1364: ADMA System Address Register Low 1109
Offset: 0x000D805C
Table 1365: Slot Interrupt Status Register 1109
Offset: 0x000D80FC
Table 1366: SD Extra Parameters Register 1110
Offset: 0x000D8100
Table 1367: FIFO Parameters Register 1110
Offset: 0x000D8104

Copyright © 2015 Marvell

Table 1368: SPI Mode and Clock and Burst Size Setup Register 1112
Offset: 0x000D8108
Table 1369: CE-ATA Register 1113
Offset: 0x000D810C
Table 1370: Drive Strength Register 1114
Offset: 0x000D8110
Table 1371: RX Configuration Register 1115
Offset: 0x000D8114
Table 1372: TX Configuration Register 1115
Offset: 0x000D8118
Table 1373: SD Host Controller Version 0 Register 1116
Offset: 0x000D81F8
Table 1374: SD Host Controller Version 1 Register 1116
Offset: 0x000D81FC
Table 1375: Mbus Port Control Register 1116
Offset: 0x000DC000
Table 1376: AXI Port Control Register 1117
Offset: 0x000DC004
Table 1377: Bridge Window Control Register (N=0–7) 1117
Offset: Windows0: 0x000DC080, Windows1: 0x000DC088, Windows2: 0x000DC090, Windows3: 0x000DC098, Windows4: 0x000DC0A0, Windows5: 0x000DC0A8, Windows6: 0x000DC0B0, Windows7: 0x000DC0B8
Table 1378: Bridge Window Base Register (N=0–7) 1118
Offset: Windows0: 0x000DC084, Windows1: 0x000DC08C, Windows2: 0x000DC094, Windows3: 0x000DC09C, Windows4: 0x000DC0A4, Windows5: 0x000DC0AC, Windows6: 0x000DC0B4, Windows7: 0x000DC0BC
Table 1379: Mbus Control Register 1118
Offset: 0x000DC0C0
Table 1380: Sync Barrier WindowN Control Register (N=0–1) 1119
Offset: SyncbWin0: 0x000DC0C4, SyncbWin1: 0x000DC0CC
Table 1381: Sync Barrier WindowN Base Register (N=0–1) 1119
Offset: SyncbWin0: 0x000DC0C8, SyncbWin1: 0x000DC0D0

A.18 Inter-Integrated Circuit (I2C) Registers 1120

Table 1383: I2C Slave Address Register (t=0–3) 1121
Offset: I2C Port0: 0x00011000, I2C Port1: 0x00011100, I2C Port2: 0x00011200, I2C Port3: 0x00011300
Table 1384: I2C Data Register (t=0–3) 1122
Offset: I2C Port0: 0x00011004, I2C Port1: 0x00011104, I2C Port2: 0x00011204, I2C Port3: 0x00011304
Table 1385: I2C Control Register (t=0–3) 1122
Offset: I2C Port0: 0x00011008, I2C Port1: 0x00011108, I2C Port2: 0x00011208, I2C Port3: 0x00011308
Table 1386: I2C Baud Rate Register (t=0–3) 1123
Offset: I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C
Table 1387: I2C Status Register (t=0–3) 1123
Offset: I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C
Table 1388: I2C Soft Reset Register (t=0–3) 1125
Offset: I2C Port0: 0x0001101C, I2C Port1: 0x0001111C, I2C Port2: 0x0001121C, I2C Port3: 0x0001131C

Copyright © 2015 Marvell

Table 1389: I2C Configuration Debug Register 1125
Offset: 0x0001108C
Table 1390: I2C Initialization Last Data Register 1125
Offset: 0x00011098
Table 1391: I2C Crossbar Port w Remap Register (w=0–15) 1125
Offset: port index0: 0x00013200, port index1: 0x00013204...port index15: 0x0001323C
Table 1392: I2C Bridge Transmit Data Low Register (t=0–3) 1126
Offset: I2C Port0: 0x000110C0, I2C Port1: 0x000111C0, I2C Port2: 0x000112C0, I2C Port3: 0x000113C0
Table 1393: I2C Bridge Transmit Data High Register (t=0–3) 1126
Offset: I2C Port0: 0x000110C4, I2C Port1: 0x000111C4, I2C Port2: 0x000112C4, I2C Port3: 0x000113C4
Table 1394: I2C Bridge Receive Data Low Register (t=0–3) 1126
Offset: I2C Port0: 0x000110C8, I2C Port1: 0x000111C8, I2C Port2: 0x000112C8, I2C Port3: 0x000113C8
Table 1395: I2C Bridge Receive Data High Register (t=0–3) 1126
Offset: I2C Port0: 0x000110CC, I2C Port1: 0x000111CC, I2C Port2: 0x000112CC, I2C Port3: 0x000113CC
Table 1396: I2C Bridge Control Register (t=0–3) 1126
Offset: I2C Port0: 0x000110D0, I2C Port1: 0x000111D0, I2C Port2: 0x000112D0, I2C Port3: 0x000113D0
Table 1397: I2C Bridge Status Register (t=0–3) 1127
Offset: I2C Port0: 0x000110D4, I2C Port1: 0x000111D4, I2C Port2: 0x000112D4, I2C Port3: 0x000113D4
Table 1398: I2C Bridge Interrupt Cause Register (t=0–3) 1128
Offset: I2C Port0: 0x000110D8, I2C Port1: 0x000111D8, I2C Port2: 0x000112D8, I2C Port3: 0x000113D8
Table 1399: I2C Bridge Interrupt Mask Register (t=0–3) 1128
Offset: I2C Port0: 0x000110DC, I2C Port1: 0x000111DC, I2C Port2: 0x000112DC, I2C Port3: 0x000113DC
Table 1400: I2C Bridge Timing Gaps Register (t=0–3) 1129
Offset: I2C Port0: 0x000110E0, I2C Port1: 0x000111E0, I2C Port2: 0x000112E0, I2C Port3: 0x000113E0

A.19 Universal Asynchronous Receiver/Transmitter (UART) Registers 1130

Table 1402: Divisor Latch Low (DLL) Register (n=0–3) 1131
Offset: UART0: 0x00012000, UART1: 0x00012100,
Table 1403: Transmit Holding (THR) Register (n=0–3) 1131
Offset: UART0: 0x00012000, UART1: 0x00012100,
Table 1404: Receive Buffer (RBR) Register (n=0–3) 1131
Offset: UART0: 0x00012000, UART1: 0x00012100,
Table 1405: Interrupt Enable (IER) Register (n=0–3) 1132
Offset: UART0: 0x00012004, UART1: 0x00012104,
Table 1406: Divisor Latch High (DLH) Register (n=0–3) 1133
Offset: UART0: 0x00012004, UART1: 0x00012104,
Table 1407: FIFO Control (FCR) Register (n=0–3) 1133
Offset: UART0: 0x00012008, UART1: 0x00012108,
Table 1408: Interrupt Identity (IIR) Register (n=0–3) 1134
Offset: UART0: 0x00012008, UART1: 0x00012108,
Table 1409: Line Control (LCR) Register (n=0–3) 1134
Offset: UART0: 0x0001200C, UART1: 0x0001210C,

Document Classification: Proprietary Information

Table 1410: Modem Control (MCR) Register (n=0–3) 1135
Offset: UART0: 0x00012010, UART1: 0x00012110,
Table 1411: Line Status (LSR) Register (n=0–3) 1136
Offset: UART0: 0x00012014, UART1: 0x00012114,
Table 1412: Modem Status (MSR) Register (n=0–3) 1137
Offset: UART0: 0x00012018, UART1: 0x00012118,
Table 1413: Scratch Pad (SCR) Register (n=0–3) 1137
Offset: UART0: 0x0001201C, UART1: 0x0001211C,
Table 1414: FIFO Access (FAR) Register (n=0–3) 1138
Offset: UART0: 0x00012070, UART1: 0x00012170,
Table 1415: UART Status (USR) Register (n=0–3) 1138
Offset: UART0: 0x0001207C, UART1: 0x0001217C,
Table 1416: Halt TX (HTX) Register (n=0–3) 1139
Offset: UART0: 0x000120A4, UART1: 0x000121A4,

A.20 General Purpose Input/Output (GPIO) Registers 1140

Table 1418: GPIO_0_31_DataOut Register 1141
Offset: 0x00018100
Table 1419: GPIO_0_31_Data Out Enable Control Register 1141
Offset: 0x00018104
Table 1420: GPIO_0_31_Blink Enable Register 1141
Offset: 0x00018108
Table 1421: GPIO_0_31_Data In Polarity Register 1141
Offset: 0x0001810C
Table 1422: GPIO_0_31_Data In Register 1142
Offset: 0x00018110
Table 1423: GPIO_0_31_Interrupt Cause Register 1142
Offset: 0x00018114
Table 1424: GPIO_0_31_Interrupt Mask Register 1142
Offset: 0x00018118
Table 1425: GPIO_0_31_Interrupt Level Mask Register 1142
Offset: 0x0001811C
Table 1426: GPIO_0_31_Blink_Counter_Select Register 1143
Offset: 0x00018120
Table 1427: GPIO_0_31_Control_Set Register 1143
Offset: 0x00018128
Table 1428: GPIO_0_31_Control_Clear Register 1143
Offset: 0x0001812C
Table 1429: GPIO_0_31_ Data_Out_Set Register 1143
Offset: 0x00018130
Table 1430: GPIO_0_31_Data_Out_Clear Register 1143
Offset: 0x00018134
Table 1431: GPIO_32_59_DataOut Register 1144
Offset: 0x00018140
Table 1432: GPIO_32_59_Data Out Enable Control Register 1144
Offset: 0x00018144
Table 1433: GPIO_32_59_Blink Enable Register 1144
Offset: 0x00018148

Copyright © 2015 Marvell

Table 1434: GPIO_32_59_Data In Polarity Register 1144
Offset: 0x0001814C
Table 1435: GPIO_32_59_Data In Register 1144
Offset: 0x00018150
Table 1436: GPIO_32_59_Interrupt Cause Register 1145
Offset: 0x00018154
Table 1437: GPIO_32_59_Interrupt Mask Register 1145
Offset: 0x00018158
Table 1438: GPIO_32_59_Interrupt Level Mask Register 1145
Offset: 0x0001815C
Table 1439: GPIO_32_59_Blink_Counter_Select Register 1146
Offset: 0x00018160
Table 1440: GPIO_32_59_Control_Set Register 1146
Offset: 0x00018168
Table 1441: GPIO_32_59_Control_Clear Register 1146
Offset: 0x0001816C
Table 1442: GPIO_32_59_Data_Out_Set Register 1146
Offset: 0x00018170
Table 1443: GPIO_32_59 Data Out Clear Register 1147
Offset: 0x00018174
Table 1444: GPIO Blink Counter A is ON Duration Register 1147
Offset: 0x000181C0
Table 1445: GPIO Blink Counter A is OFF Duration Register 1147
Offset: 0x000181C4
Table 1446: GPIO Blink Counter B is ON Duration Register 1147
Offset: 0x000181C8
Table 1447: GPIO Blink Counter B is OFF Duration Register 1147
Offset: 0x000181CC
Table 1448: CPU<m>_GPIO_0_31_Interrupt Cause Register (m=0–1) 1148
Offset: CPU_Num0: 0x00018800, CPU_Num1: 0x00018804
Table 1449: CPU<m>_GPIO_0_31_Interrupt Mask Register (m=0–1) 1148
Offset: CPU_Num0: 0x00018810, CPU_Num1: 0x00018814
Table 1450: CPU<m>_GPIO_0_31_Interrupt Level Mask Register (m=0–1) 1148
Offset: CPU_Num0: 0x00018820, CPU_Num1: 0x00018824
Table 1451: CPU<m>_GPIO_32_59_Interrupt Cause Register (m=0–1) 1148
Offset: CPU_Num0: 0x00018840, CPU_Num1: 0x00018844
Table 1452: CPU<m>_GPIO_32_59_Interrupt Mask Register (m=0–1) 1149
Offset: CPU_Num0: 0x00018850, CPU_Num1: 0x00018854
Table 1453: CPU<m>_GPIO_32_59_Interrupt Level Mask Register (m=0–1) 1149
Offset: CPU_Num0: 0x00018860, CPU_Num1: 0x00018864

A.21 Real-Time Clock (RTC) Registers 1150

Table 1455: RTC Status Register 1150
Offset: 0x000A3800
Table 1456: RTC Interrupt 1 Configuration Register 1150
Offset: 0x000A3804
Table 1457: RTC Interrupt 2 Configuration Register 1151
Offset: 0x000A3808

Copyright © 2015 Marvell

Table 1458: RTC Time Register 1151
Offset: 0x000A380C
Table 1459: RTC Alarm 1 Register 1152
Offset: 0x000A3810
Table 1460: RTC Alarm 2 Register 1152
Offset: 0x000A3814
Table 1461: RTC Clock Correction Register 1152
Offset: 0x000A3818
Table 1462: RTC Test Configuration Register 1152
Offset: 0x000A381C

A.22 Power Management Unit Registers 1153

Table 1464: PMU Power Domain Delay Select Register 1153
Offset: 0x0001C010
Table 1465: PMU Power Up Delay 0 Register 1153
Offset: 0x0001C014
Table 1466: PMU Power Up Delay 1 Register 1154
Offset: 0x0001C018
Table 1467: PMU DFS Control 1 Register 1154
Offset: 0x0001C054
Table 1468: PMU Activate I/F Control Register 1154
Offset: 0x0001C090

A.23 Common PHY Registers 1155

Table 1470: Common PHY<n> Configuration 1 Register (n=0–5) 1156
Offset: PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8
Table 1471: Common PHY<n> Configuration 2 Register (n=0–5) 1160
Offset: PHY_NUM0: 0x00018304, PHY_NUM1: 0x0001832C, PHY_NUM2: 0x00018354, PHY_NUM3: 0x0001837C, PHY_NUM4: 0x000183A4, PHY_NUM5: 0x000183CC
Table 1472: Common PHY<n> Configuration 4 Register (n=0–5) 1161
Offset: PHY_NUM0: 0x0001830C, PHY_NUM1: 0x00018334, PHY_NUM2: 0x0001835C, PHY_NUM3: 0x00018384, PHY_NUM4: 0x000183AC, PHY_NUM5: 0x000183D4
Table 1473: Common PHY<n> Configuration 5 Register (n=0–5) 1162
Offset: PHY_NUM0: 0x00018310, PHY_NUM1: 0x00018338, PHY_NUM2: 0x00018360, PHY_NUM3: 0x00018388, PHY_NUM4: 0x000183B0, PHY_NUM5: 0x000183D8
Table 1474: Common PHY<n> Configuration 6 Register (n=0–5) 1164
Offset: PHY_NUM0: 0x00018314, PHY_NUM1: 0x0001833C, PHY_NUM2: 0x00018364, PHY_NUM3: 0x0001838C, PHY_NUM4: 0x000183B4, PHY_NUM5: 0x000183DC
Table 1475: Common PHY<n> Status 1 Register (n=0–5) 1164
Offset: PHY_NUM0: 0x00018318, PHY_NUM1: 0x00018340, PHY_NUM2: 0x00018368, PHY_NUM3: 0x00018390, PHY_NUM4: 0x000183B8, PHY_NUM5: 0x000183E0
Table 1476: Common PHY<n> Status 2 Register (n=0–5) 1166
Offset: PHY_NUM0: 0x0001831C, PHY_NUM1: 0x00018344, PHY_NUM2: 0x0001836C, PHY_NUM3: 0x00018394, PHY_NUM4: 0x000183BC, PHY_NUM5: 0x000183E4
Table 1477: Common PHY<n> Status 3 Register (n=0–5) 1167
Offset: PHY_NUM0: 0x00018320, PHY_NUM1: 0x00018348, PHY_NUM2: 0x00018370, PHY_NUM3: 0x00018398, PHY_NUM4: 0x000183C0, PHY_NUM5: 0x000183E8
Table 1478: Common PHY<n> Status 4 Register (n=0–5) 1168
Offset: PHY_NUM0: 0x00018324, PHY_NUM1: 0x0001834C, PHY_NUM2: 0x00018374, PHY_

NUM3: 0x0001839C, PHY_NUM4: 0x000183C4, PHY_NUM5: 0x000183EC
Table 1479: Common PHYs Selectors Register 1168
Offset: 0x000183FC

A.24 Common PHY and PIPE Interface Registers 1170
Table 1481: Chip ID Register (n=0–6) 1204
Offset: Port0: 0x000A0000, Port1: 0x000A0800, Port2: 0x000A1000, Port3: 0x000A1800, Port4: 0x000A2000, Port5: 0x000A2800,
Table 1482: Power and PLL Control Register (n=0–6) 1205
Offset: Port0: 0x000A0004, Port1: 0x000A0804, Port2: 0x000A1004, Port3: 0x000A1804, Port4: 0x000A2004, Port5: 0x000A2804,
Table 1483: KVCO Calibration Control Register (n=0–6) 1208
Offset: Port0: 0x000A0008, Port1: 0x000A0808, Port2: 0x000A1008, Port3: 0x000A1808, Port4: 0x000A2008, Port5: 0x000A2808,
Table 1484: Impedance Calibration Control Register (n=0–6) 1210
Offset: Port0: 0x000A000C, Port1: 0x000A080C, Port2: 0x000A100C, Port3: 0x000A180C, Port4: 0x000A200C, Port5: 0x000A280C,
Table 1485: Impedance and TX SSC Control Register (n=0–6) 1210
Offset: Port0: 0x000A0010, Port1: 0x000A0810, Port2: 0x000A1010, Port3: 0x000A1810, Port4: 0x000A2010, Port5: 0x000A2810,
Table 1486: Transmitter Frequency Offset Register (n=0–6) 1211
Offset: Port0: 0x000A0014, Port1: 0x000A0814, Port2: 0x000A1014, Port3: 0x000A1814, Port4: 0x000A2014, Port5: 0x000A2814,
Table 1487: Squelch and FFE Setting Register (n=0–6) 1211
Offset: Port0: 0x000A0018, Port1: 0x000A0818, Port2: 0x000A1018, Port3: 0x000A1818, Port4: 0x000A2018, Port5: 0x000A2818,
Table 1488: DFE Resolution Control Register (n=0–6) 1212
Offset: Port0: 0x000A001C, Port1: 0x000A081C, Port2: 0x000A101C, Port3: 0x000A181C, Port4: 0x000A201C, Port5: 0x000A281C,
Table 1489: DFE Update Control Register (n=0–6) 1213
Offset: Port0: 0x000A0020, Port1: 0x000A0820, Port2: 0x000A1020, Port3: 0x000A1820, Port4: 0x000A2020, Port5: 0x000A2820,
Table 1490: DFE F0-F2 Coefficient Control Register (n=0–6) 1214
Offset: Port0: 0x000A0024, Port1: 0x000A0824, Port2: 0x000A1024, Port3: 0x000A1824, Port4: 0x000A2024, Port5: 0x000A2824,
Table 1491: DFE F3-F5 Coefficient Control Register (n=0–6) 1214
Offset: Port0: 0x000A0028, Port1: 0x000A0828, Port2: 0x000A1028, Port3: 0x000A1828, Port4: 0x000A2028, Port5: 0x000A2828,
Table 1492: Adapted DFE Coefficient 0 Register (n=0–6) 1215
Offset: Port0: 0x000A002C, Port1: 0x000A082C, Port2: 0x000A102C, Port3: 0x000A182C, Port4: 0x000A202C, Port5: 0x000A282C,
Table 1493: Adapted DFE Coefficient 1 Register (n=0–6) 1215
Offset: Port0: 0x000A0030, Port1: 0x000A0830, Port2: 0x000A1030, Port3: 0x000A1830, Port4: 0x000A2030, Port5: 0x000A2830,
Table 1494: Generation 1 Setting 0 Register (n=0–6) 1216
Offset: Port0: 0x000A0034, Port1: 0x000A0834, Port2: 0x000A1034, Port3: 0x000A1834, Port4: 0x000A2034, Port5: 0x000A2834,
Table 1495: Generation 1 Setting 1 Register (n=0–6) 1216
Offset: Port0: 0x000A0038, Port1: 0x000A0838, Port2: 0x000A1038, Port3: 0x000A1838, Port4: 0x000A2038, Port5: 0x000A2838,

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1496: Generation 2 Setting 0 Register (n=0–6) 1218
Offset: Port0: 0x000A003C, Port1: 0x000A083C, Port2: 0x000A103C, Port3: 0x000A183C, Port4: 0x000A203C, Port5: 0x000A283C,
Table 1497: Generation 2 Setting 1 Register (n=0–6) 1219
Offset: Port0: 0x000A0040, Port1: 0x000A0840, Port2: 0x000A1040, Port3: 0x000A1840, Port4: 0x000A2040, Port5: 0x000A2840,
Table 1498: Generation 3 Setting 0 Register (n=0–6) 1220
Offset: Port0: 0x000A0044, Port1: 0x000A0844, Port2: 0x000A1044, Port3: 0x000A1844, Port4: 0x000A2044, Port5: 0x000A2844,
Table 1499: Generation 3 Setting 1 Register (n=0–6) 1221
Offset: Port0: 0x000A0048, Port1: 0x000A0848, Port2: 0x000A1048, Port3: 0x000A1848, Port4: 0x000A2048, Port5: 0x000A2848,
Table 1500: Generation 4 Setting 0 Register (n=0–6) 1223
Offset: Port0: 0x000A004C, Port1: 0x000A084C, Port2: 0x000A104C, Port3: 0x000A184C, Port4: 0x000A204C, Port5: 0x000A284C,
Table 1501: Generation 4 Setting 1 Register (n=0–6) 1223
Offset: Port0: 0x000A0050, Port1: 0x000A0850, Port2: 0x000A1050, Port3: 0x000A1850, Port4: 0x000A2050, Port5: 0x000A2850,
Table 1502: PHY Test Control 0 Register (n=0–6) 1225
Offset: Port0: 0x000A0054, Port1: 0x000A0854, Port2: 0x000A1054, Port3: 0x000A1854, Port4: 0x000A2054, Port5: 0x000A2854,
Table 1503: PHY Test Data 0 Register (n=0–6) 1226
Offset: Port0: 0x000A0058, Port1: 0x000A0858, Port2: 0x000A1058, Port3: 0x000A1858, Port4: 0x000A2058, Port5: 0x000A2858,
Table 1504: PHY Test Data 1 Register (n=0–6) 1227
Offset: Port0: 0x000A005C, Port1: 0x000A085C, Port2: 0x000A105C, Port3: 0x000A185C, Port4: 0x000A205C, Port5: 0x000A285C,
Table 1505: PHY Test Data 2 Register (n=0–6) 1227
Offset: Port0: 0x000A0060, Port1: 0x000A0860, Port2: 0x000A1060, Port3: 0x000A1860, Port4: 0x000A2060, Port5: 0x000A2860,
Table 1506: PHY Test Data 3 Register (n=0–6) 1228
Offset: Port0: 0x000A0064, Port1: 0x000A0864, Port2: 0x000A1064, Port3: 0x000A1864, Port4: 0x000A2064, Port5: 0x000A2864,
Table 1507: PHY Test Data 4 Register (n=0–6) 1228
Offset: Port0: 0x000A0068, Port1: 0x000A0868, Port2: 0x000A1068, Port3: 0x000A1868, Port4: 0x000A2068, Port5: 0x000A2868,
Table 1508: PHY Test Data 5 Register (n=0–6) 1229
Offset: Port0: 0x000A006C, Port1: 0x000A086C, Port2: 0x000A106C, Port3: 0x000A186C, Port4: 0x000A206C, Port5: 0x000A286C,
Table 1509: PHY Test PRBS Counter 0 Register (n=0–6) 1229
Offset: Port0: 0x000A0070, Port1: 0x000A0870, Port2: 0x000A1070, Port3: 0x000A1870, Port4: 0x000A2070, Port5: 0x000A2870,
Table 1510: PHY Test PRBS Counter 1 Register (n=0–6) 1229
Offset: Port0: 0x000A0074, Port1: 0x000A0874, Port2: 0x000A1074, Port3: 0x000A1874, Port4: 0x000A2074, Port5: 0x000A2874,
Table 1511: PHY Test PRBS Counter 2 Register (n=0–6) 1229
Offset: Port0: 0x000A0078, Port1: 0x000A0878, Port2: 0x000A1078, Port3: 0x000A1878, Port4: 0x000A2078, Port5: 0x000A2878,
Table 1512: PHY Test PRBS Error Counter 0 Register (n=0–6) 1230
Offset: Port0: 0x000A007C, Port1: 0x000A087C, Port2: 0x000A107C, Port3: 0x000A187C, Port4: 0x000A207C, Port5: 0x000A287C,

Table 1513: PHY Test PRBS Error Counter 1 Register (n=0–6) 1230
Offset: Port0: 0x000A0080, Port1: 0x000A0880, Port2: 0x000A1080, Port3: 0x000A1880, Port4: 0x000A2080, Port5: 0x000A2880,

Table 1514: PHY Test OOB 0 Register (n=0–6) 1230
Offset: Port0: 0x000A0084, Port1: 0x000A0884, Port2: 0x000A1084, Port3: 0x000A1884, Port4: 0x000A2084, Port5: 0x000A2884,

Table 1515: PHY Test OOB Register (n=0–6) 1230
Offset: Port0: 0x000A0088, Port1: 0x000A0888, Port2: 0x000A1088, Port3: 0x000A1888, Port4: 0x000A2088, Port5: 0x000A2888,

Table 1516: Digital Loopback Enable Register (n=0–6) 1231
Offset: Port0: 0x000A008C, Port1: 0x000A088C, Port2: 0x000A108C, Port3: 0x000A188C, Port4: 0x000A208C, Port5: 0x000A288C,

Table 1517: Sync Pattern Register (n=0–6) 1232
Offset: Port0: 0x000A0090, Port1: 0x000A0890, Port2: 0x000A1090, Port3: 0x000A1890, Port4: 0x000A2090, Port5: 0x000A2890,

Table 1518: Sync Mask and Maximal Generation Setting Register (n=0–6) 1233
Offset: Port0: 0x000A0094, Port1: 0x000A0894, Port2: 0x000A1094, Port3: 0x000A1894, Port4: 0x000A2094, Port5: 0x000A2894,

Table 1519: PHY Isolation Mode Control Register (n=0–6) 1234
Offset: Port0: 0x000A0098, Port1: 0x000A0898, Port2: 0x000A1098, Port3: 0x000A1898, Port4: 0x000A2098, Port5: 0x000A2898,

Table 1520: EOM Circuit Control Register (n=0–6) 1236
Offset: Port0: 0x000A009C, Port1: 0x000A089C, Port2: 0x000A109C, Port3: 0x000A189C, Port4: 0x000A209C, Port5: 0x000A289C,

Table 1521: EOM Set 1 Control 0 Register (n=0–6) 1237
Offset: Port0: 0x000A00A0, Port1: 0x000A08A0, Port2: 0x000A10A0, Port3: 0x000A18A0, Port4: 0x000A20A0, Port5: 0x000A28A0,

Table 1522: EOM Set 1 Control 1 Register (n=0–6) 1239
Offset: Port0: 0x000A00A4, Port1: 0x000A08A4, Port2: 0x000A10A4, Port3: 0x000A18A4, Port4: 0x000A20A4, Port5: 0x000A28A4,

Table 1523: EOM Set 1 Population Counter 0 Register (n=0–6) 1239
Offset: Port0: 0x000A00A8, Port1: 0x000A08A8, Port2: 0x000A10A8, Port3: 0x000A18A8, Port4: 0x000A20A8, Port5: 0x000A28A8,

Table 1524: EOM Set 1 Population Counter 1 Register (n=0–6) 1240
Offset: Port0: 0x000A00AC, Port1: 0x000A08AC, Port2: 0x000A10AC, Port3: 0x000A18AC, Port4: 0x000A20AC, Port5: 0x000A28AC,

Table 1525: EOM Set 1 Population Counter 2 Register (n=0–6) 1240
Offset: Port0: 0x000A00B0, Port1: 0x000A08B0, Port2: 0x000A10B0, Port3: 0x000A18B0, Port4: 0x000A20B0, Port5: 0x000A28B0,

Table 1526: EOM Set 1 Population Counter 3 Register (n=0–6) 1240
Offset: Port0: 0x000A00B4, Port1: 0x000A08B4, Port2: 0x000A10B4, Port3: 0x000A18B4, Port4: 0x000A20B4, Port5: 0x000A28B4,

Table 1527: EOM Set 1 Threshold Counter 0 Register (n=0–6) 1240
Offset: Port0: 0x000A00B8, Port1: 0x000A08B8, Port2: 0x000A10B8, Port3: 0x000A18B8, Port4: 0x000A20B8, Port5: 0x000A28B8,

Table 1528: EOM Set 1 Threshold Counter 1 Register (n=0–6) 1241
Offset: Port0: 0x000A00BC, Port1: 0x000A08BC, Port2: 0x000A10BC, Port3: 0x000A18BC, Port4: 0x000A20BC, Port5: 0x000A28BC,

Table 1529: EOM Set 1 Threshold Counter 2 Register (n=0–6) 1241
Offset: Port0: 0x000A00C0, Port1: 0x000A08C0, Port2: 0x000A10C0, Port3: 0x000A18C0, Port4: 0x000A20C0, Port5: 0x000A28C0,

Document Classification: Proprietary Information

Table 1530: EOM Set 1 Threshold Counter 3 Register (n=0–6) 1242
Offset: Port0: 0x000A00C4, Port1: 0x000A08C4, Port2: 0x000A10C4, Port3: 0x000A18C4, Port4: 0x000A20C4, Port5: 0x000A28C4,
Table 1531: EOM Set 1 Sample Counter 0 Register (n=0–6) 1242
Offset: Port0: 0x000A00C8, Port1: 0x000A08C8, Port2: 0x000A10C8, Port3: 0x000A18C8, Port4: 0x000A20C8, Port5: 0x000A28C8,
Table 1532: EOM Set 1 Sample Counter 1 Register (n=0–6) 1242
Offset: Port0: 0x000A00CC, Port1: 0x000A08CC, Port2: 0x000A10CC, Port3: 0x000A18CC, Port4: 0x000A20CC, Port5: 0x000A28CC,
Table 1533: EOM Set 1 Sample Counter 2 Register (n=0–6) 1242
Offset: Port0: 0x000A00D0, Port1: 0x000A08D0, Port2: 0x000A10D0, Port3: 0x000A18D0, Port4: 0x000A20D0, Port5: 0x000A28D0,
Table 1534: EOM Set 1 Sample Counter 3 Register (n=0–6) 1243
Offset: Port0: 0x000A00D4, Port1: 0x000A08D4, Port2: 0x000A10D4, Port3: 0x000A18D4, Port4: 0x000A20D4, Port5: 0x000A28D4,
Table 1535: EOM Set 1 Error Counter 0 Register (n=0–6) 1243
Offset: Port0: 0x000A00D8, Port1: 0x000A08D8, Port2: 0x000A10D8, Port3: 0x000A18D8, Port4: 0x000A20D8, Port5: 0x000A28D8,
Table 1536: EOM Set 1 Error Counter 1 Register (n=0–6) 1243
Offset: Port0: 0x000A00DC, Port1: 0x000A08DC, Port2: 0x000A10DC, Port3: 0x000A18DC, Port4: 0x000A20DC, Port5: 0x000A28DC,
Table 1537: EOM Set 1 Error Counter 2 Register (n=0–6) 1243
Offset: Port0: 0x000A00E0, Port1: 0x000A08E0, Port2: 0x000A10E0, Port3: 0x000A18E0, Port4: 0x000A20E0, Port5: 0x000A28E0,
Table 1538: EOM Set 1 Error Counter 3 Register (n=0–6) 1244
Offset: Port0: 0x000A00E4, Port1: 0x000A08E4, Port2: 0x000A10E4, Port3: 0x000A18E4, Port4: 0x000A20E4, Port5: 0x000A28E4,
Table 1539: External Digital Reserved 0 Register (n=0–6) 1244
Offset: Port0: 0x000A00E8, Port1: 0x000A08E8, Port2: 0x000A10E8, Port3: 0x000A18E8, Port4: 0x000A20E8, Port5: 0x000A28E8,
Table 1540: DFE Step Size 0 Register (n=0–6) 1244
Offset: Port0: 0x000A00EC, Port1: 0x000A08EC, Port2: 0x000A10EC, Port3: 0x000A18EC, Port4: 0x000A20EC, Port5: 0x000A28EC,
Table 1541: DFE Step Size 1 Register (n=0–6) 1246
Offset: Port0: 0x000A00F0, Port1: 0x000A08F0, Port2: 0x000A10F0, Port3: 0x000A18F0, Port4: 0x000A20F0, Port5: 0x000A28F0,
Table 1542: Generation 1 Setting 2 Register (n=0–6) 1247
Offset: Port0: 0x000A00F4, Port1: 0x000A08F4, Port2: 0x000A10F4, Port3: 0x000A18F4, Port4: 0x000A20F4, Port5: 0x000A28F4,
Table 1543: Generation 2 Setting 2 Register (n=0–6) 1248
Offset: Port0: 0x000A00F8, Port1: 0x000A08F8, Port2: 0x000A10F8, Port3: 0x000A18F8, Port4: 0x000A20F8, Port5: 0x000A28F8,
Table 1544: Generation 3 Setting 2 Register (n=0–6) 1249
Offset: Port0: 0x000A00FC, Port1: 0x000A08FC, Port2: 0x000A10FC, Port3: 0x000A18FC, Port4: 0x000A20FC, Port5: 0x000A28FC,
Table 1545: Generation 4 Setting 2 Register (n=0–6) 1249
Offset: Port0: 0x000A0100, Port1: 0x000A0900, Port2: 0x000A1100, Port3: 0x000A1900, Port4: 0x000A2100, Port5: 0x000A2900,
Table 1546: VTHIMPCAL Value Control Register (n=0–6) 1250
Offset: Port0: 0x000A0104, Port1: 0x000A0904, Port2: 0x000A1104, Port3: 0x000A1904, Port4: 0x000A2104, Port5: 0x000A2904,

Table 1547: VDD Calibration Control 0 Register (n=0–6) 1251
Offset: Port0: 0x000A0108, Port1: 0x000A0908, Port2: 0x000A1108, Port3: 0x000A1908, Port4: 0x000A2108, Port5: 0x000A2908,

Table 1548: VDD Calibration Control 1 Register (n=0–6) 1251
Offset: Port0: 0x000A010C, Port1: 0x000A090C, Port2: 0x000A110C, Port3: 0x000A190C, Port4: 0x000A210C, Port5: 0x000A290C,

Table 1549: VDD Calibration Control 2 Register (n=0–6) 1251
Offset: Port0: 0x000A0110, Port1: 0x000A0910, Port2: 0x000A1110, Port3: 0x000A1910, Port4: 0x000A2110, Port5: 0x000A2910,

Table 1550: VDD Calibration Control 3 Register (n=0–6) 1251
Offset: Port0: 0x000A0114, Port1: 0x000A0914, Port2: 0x000A1114, Port3: 0x000A1914, Port4: 0x000A2114, Port5: 0x000A2914,

Table 1551: Analog Reserved 1 Register (n=0–6) 1252
Offset: Port0: 0x000A0118, Port1: 0x000A0918, Port2: 0x000A1118, Port3: 0x000A1918, Port4: 0x000A2118, Port5: 0x000A2918,

Table 1552: Analog Reserved 2 Register (n=0–6) 1252
Offset: Port0: 0x000A011C, Port1: 0x000A091C, Port2: 0x000A111C, Port3: 0x000A191C, Port4: 0x000A211C, Port5: 0x000A291C,

Table 1553: PCIe/USB3 Control Register (n=0–6) 1252
Offset: Port0: 0x000A0120, Port1: 0x000A0920, Port2: 0x000A1120, Port3: 0x000A1920, Port4: 0x000A2120, Port5: 0x000A2920,

Table 1554: Lane Alignment Control Register (n=0–6) 1252
Offset: Port0: 0x000A0124, Port1: 0x000A0924, Port2: 0x000A1124, Port3: 0x000A1924, Port4: 0x000A2124, Port5: 0x000A2924,

Table 1555: Analog Test Point Selection Register (n=0–6) 1253
Offset: Port0: 0x000A0128, Port1: 0x000A0928, Port2: 0x000A1128, Port3: 0x000A1928, Port4: 0x000A2128, Port5: 0x000A2928,

Table 1556: 150M Clock Control 0 Register (n=0–6) 1253
Offset: Port0: 0x000A012C, Port1: 0x000A092C, Port2: 0x000A112C, Port3: 0x000A192C, Port4: 0x000A212C, Port5: 0x000A292C,

Table 1557: 150M Clock Control 1 Register (n=0–6) 1253
Offset: Port0: 0x000A0130, Port1: 0x000A0930, Port2: 0x000A1130, Port3: 0x000A1930, Port4: 0x000A2130, Port5: 0x000A2930,

Table 1558: Regulator Output Control 0 Register (n=0–6) 1253
Offset: Port0: 0x000A0134, Port1: 0x000A0934, Port2: 0x000A1134, Port3: 0x000A1934, Port4: 0x000A2134, Port5: 0x000A2934,

Table 1559: Regulator Output Control 1 Register (n=0–6) 1254
Offset: Port0: 0x000A0138, Port1: 0x000A0938, Port2: 0x000A1138, Port3: 0x000A1938, Port4: 0x000A2138, Port5: 0x000A2938,

Table 1560: Miscellaneous Control 0 Register (n=0–6) 1254
Offset: Port0: 0x000A013C, Port1: 0x000A093C, Port2: 0x000A113C, Port3: 0x000A193C, Port4: 0x000A213C, Port5: 0x000A293C,

Table 1561: RX Control 1 Register (n=0–6) 1256
Offset: Port0: 0x000A0140, Port1: 0x000A0940, Port2: 0x000A1140, Port3: 0x000A1940, Port4: 0x000A2140, Port5: 0x000A2940,

Table 1562: Interface and Digital Reference Clock Control Register (n=0–6) 1256
Offset: Port0: 0x000A0144, Port1: 0x000A0944, Port2: 0x000A1144, Port3: 0x000A1944, Port4: 0x000A2144, Port5: 0x000A2944,

Table 1563: Power Control Register (n=0–6) 1257
Offset: Port0: 0x000A0148, Port1: 0x000A0948, Port2: 0x000A1148, Port3: 0x000A1948, Port4: 0x000A2148, Port5: 0x000A2948,

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1564: Feedback Divider Register (n=0–6) 1257
Offset: Port0: 0x000A014C, Port1: 0x000A094C, Port2: 0x000A114C, Port3: 0x000A194C, Port4: 0x000A214C, Port5: 0x000A294C,

Table 1565: Forward Divider Register (n=0–6) 1257
Offset: Port0: 0x000A0150, Port1: 0x000A0950, Port2: 0x000A1150, Port3: 0x000A1950, Port4: 0x000A2150, Port5: 0x000A2950,

Table 1566: Transmitter/Receiver Speed Divider Force Register (n=0–6) 1257
Offset: Port0: 0x000A0154, Port1: 0x000A0954, Port2: 0x000A1154, Port3: 0x000A1954, Port4: 0x000A2154, Port5: 0x000A2954,

Table 1567: PLL Bandwidth Adjust Register (n=0–6) 1258
Offset: Port0: 0x000A0158, Port1: 0x000A0958, Port2: 0x000A1158, Port3: 0x000A1958, Port4: 0x000A2158, Port5: 0x000A2958,

Table 1568: Calibration Enable Control Register (n=0–6) 1258
Offset: Port0: 0x000A015C, Port1: 0x000A095C, Port2: 0x000A115C, Port3: 0x000A195C, Port4: 0x000A215C, Port5: 0x000A295C,

Table 1569: TX DCC Control Register (n=0–6) 1259
Offset: Port0: 0x000A0160, Port1: 0x000A0960, Port2: 0x000A1160, Port3: 0x000A1960, Port4: 0x000A2160, Port5: 0x000A2960,

Table 1570: RX DCC Edge and Center Control Register (n=0–6) 1259
Offset: Port0: 0x000A0164, Port1: 0x000A0964, Port2: 0x000A1164, Port3: 0x000A1964, Port4: 0x000A2164, Port5: 0x000A2964,

Table 1571: RX CLK Align 90 and TX Idle Calibration Control Register (n=0–6) 1259
Offset: Port0: 0x000A0168, Port1: 0x000A0968, Port2: 0x000A1168, Port3: 0x000A1968, Port4: 0x000A2168, Port5: 0x000A2968,

Table 1572: Sampler and Process Calibration Control Register (n=0–6) 1260
Offset: Port0: 0x000A016C, Port1: 0x000A096C, Port2: 0x000A116C, Port3: 0x000A196C, Port4: 0x000A216C, Port5: 0x000A296C,

Table 1573: Process Threshold Setting Register (n=0–6) 1260
Offset: Port0: 0x000A0170, Port1: 0x000A0970, Port2: 0x000A1170, Port3: 0x000A1970, Port4: 0x000A2170, Port5: 0x000A2970,

Table 1574: Transmitter Slew Rate Control Register (n=0–6) 1260
Offset: Port0: 0x000A0174, Port1: 0x000A0974, Port2: 0x000A1174, Port3: 0x000A1974, Port4: 0x000A2174, Port5: 0x000A2974,

Table 1575: Transmitter Amplitude and Emphasis Control Register (n=0–6) 1261
Offset: Port0: 0x000A0178, Port1: 0x000A0978, Port2: 0x000A1178, Port3: 0x000A1978, Port4: 0x000A2178, Port5: 0x000A2978,

Table 1576: SSC Control Register (n=0–6) 1261
Offset: Port0: 0x000A017C, Port1: 0x000A097C, Port2: 0x000A117C, Port3: 0x000A197C, Port4: 0x000A217C, Port5: 0x000A297C,

Table 1577: SSC Step Register (n=0–6) 1261
Offset: Port0: 0x000A0180, Port1: 0x000A0980, Port2: 0x000A1180, Port3: 0x000A1980, Port4: 0x000A2180, Port5: 0x000A2980,

Table 1578: DTL Control Register (n=0–6) 1262
Offset: Port0: 0x000A0184, Port1: 0x000A0984, Port2: 0x000A1184, Port3: 0x000A1984, Port4: 0x000A2184, Port5: 0x000A2984,

Table 1579: Offset Phase Control Register (n=0–6) 1262
Offset: Port0: 0x000A0188, Port1: 0x000A0988, Port2: 0x000A1188, Port3: 0x000A1988, Port4: 0x000A2188, Port5: 0x000A2988,

Table 1580: Receiver Frequency Offset Extraction Register (n=0–6) 1262
Offset: Port0: 0x000A018C, Port1: 0x000A098C, Port2: 0x000A118C, Port3: 0x000A198C, Port4: 0x000A218C, Port5: 0x000A298C,

Copyright © 2015 Marvell

Table 1581: Receiver Frequency Offset Read Register (n=0–6) 1262
Offset: Port0: 0x000A0190, Port1: 0x000A0990, Port2: 0x000A1190, Port3: 0x000A1990, Port4: 0x000A2190, Port5: 0x000A2990,
Table 1582: Receiver Offset Control Register (n=0–6) 1263
Offset: Port0: 0x000A0194, Port1: 0x000A0994, Port2: 0x000A1194, Port3: 0x000A1994, Port4: 0x000A2194, Port5: 0x000A2994,
Table 1583: Receiver Clamping Control Register (n=0–6) 1263
Offset: Port0: 0x000A0198, Port1: 0x000A0998, Port2: 0x000A1198, Port3: 0x000A1998, Port4: 0x000A2198, Port5: 0x000A2998,
Table 1584: DFE Miscellaneous Control Register (n=0–6) 1263
Offset: Port0: 0x000A019C, Port1: 0x000A099C, Port2: 0x000A119C, Port3: 0x000A199C, Port4: 0x000A219C, Port5: 0x000A299C,
Table 1585: Data Phase Offset Control Register (n=0–6) 1264
Offset: Port0: 0x000A01A0, Port1: 0x000A09A0, Port2: 0x000A11A0, Port3: 0x000A19A0, Port4: 0x000A21A0, Port5: 0x000A29A0,
Table 1586: Adapted FFE Capacitor/Counter Control Register (n=0–6) 1264
Offset: Port0: 0x000A01A4, Port1: 0x000A09A4, Port2: 0x000A11A4, Port3: 0x000A19A4, Port4: 0x000A21A4, Port5: 0x000A29A4,
Table 1587: Error Counter Constraint Control Register (n=0–6) 1264
Offset: Port0: 0x000A01A8, Port1: 0x000A09A8, Port2: 0x000A11A8, Port3: 0x000A19A8, Port4: 0x000A21A8, Port5: 0x000A29A8,
Table 1588: DFE Timer Register (n=0–6) 1265
Offset: Port0: 0x000A01AC, Port1: 0x000A09AC, Port2: 0x000A11AC, Port3: 0x000A19AC, Port4: 0x000A21AC, Port5: 0x000A29AC,
Table 1589: Eye Threshold Mode Selection Register (n=0–6) 1266
Offset: Port0: 0x000A01B0, Port1: 0x000A09B0, Port2: 0x000A11B0, Port3: 0x000A19B0, Port4: 0x000A21B0, Port5: 0x000A29B0,
Table 1590: RX SSC Extraction Register (n=0–6) 1266
Offset: Port0: 0x000A01B4, Port1: 0x000A09B4, Port2: 0x000A11B4, Port3: 0x000A19B4, Port4: 0x000A21B4, Port5: 0x000A29B4,
Table 1591: Digital Test Bus Configuration Register (n=0–6) 1266
Offset: Port0: 0x000A01B8, Port1: 0x000A09B8, Port2: 0x000A11B8, Port3: 0x000A19B8, Port4: 0x000A21B8, Port5: 0x000A29B8,
Table 1592: External Squelch Calibration Control Register (n=0–6) 1267
Offset: Port0: 0x000A01BC, Port1: 0x000A09BC, Port2: 0x000A11BC, Port3: 0x000A19BC, Port4: 0x000A21BC, Port5: 0x000A29BC,
Table 1593: PLL Calibration Control 0 Register (n=0–6) 1267
Offset: Port0: 0x000A01C0, Port1: 0x000A09C0, Port2: 0x000A11C0, Port3: 0x000A19C0, Port4: 0x000A21C0, Port5: 0x000A29C0,
Table 1594: PLL Calibration Control 1 Register (n=0–6) 1267
Offset: Port0: 0x000A01C4, Port1: 0x000A09C4, Port2: 0x000A11C4, Port3: 0x000A19C4, Port4: 0x000A21C4, Port5: 0x000A29C4,
Table 1595: Squelch Glitch Filter Control Register (n=0–6) 1268
Offset: Port0: 0x000A01C8, Port1: 0x000A09C8, Port2: 0x000A11C8, Port3: 0x000A19C8, Port4: 0x000A21C8, Port5: 0x000A29C8,
Table 1596: Miscellaneous Control 1 Register (n=0–6) 1268
Offset: Port0: 0x000A01CC, Port1: 0x000A09CC, Port2: 0x000A11CC, Port3: 0x000A19CC, Port4: 0x000A21CC, Port5: 0x000A29CC,
Table 1597: Miscellaneous Control 1 Register (n=0–6) 1269
Offset: Port0: 0x000A01D0, Port1: 0x000A09D0, Port2: 0x000A11D0, Port3: 0x000A19D0, Port4: 0x000A21D0, Port5: 0x000A29D0,

Copyright © 2015 Marvell

Table 1598: Squelch Calibration Register (n=0–6) 1269
Offset: Port0: 0x000A01D4, Port1: 0x000A09D4, Port2: 0x000A11D4, Port3: 0x000A19D4, Port4: 0x000A21D4, Port5: 0x000A29D4,

Table 1599: Reg0p9 Control Register (n=0–6) 1270
Offset: Port0: 0x000A01D8, Port1: 0x000A09D8, Port2: 0x000A11D8, Port3: 0x000A19D8, Port4: 0x000A21D8, Port5: 0x000A29D8,

Table 1600: Clk150M Control 2 Register (n=0–6) 1270
Offset: Port0: 0x000A01DC, Port1: 0x000A09DC, Port2: 0x000A11DC, Port3: 0x000A19DC, Port4: 0x000A21DC, Port5: 0x000A29DC,

Table 1601: Cal_reg19 Register (n=0–6) 1271
Offset: Port0: 0x000A01E0, Port1: 0x000A09E0, Port2: 0x000A11E0, Port3: 0x000A19E0, Port4: 0x000A21E0, Port5: 0x000A29E0,

Table 1602: Impedance Calibration Result Register (n=0–6) 1271
Offset: Port0: 0x000A01E4, Port1: 0x000A09E4, Port2: 0x000A11E4, Port3: 0x000A19E4, Port4: 0x000A21E4, Port5: 0x000A29E4,

Table 1603: Impedance Calibration Result Register (n=0–6) 1271
Offset: Port0: 0x000A01E8, Port1: 0x000A09E8, Port2: 0x000A11E8, Port3: 0x000A19E8, Port4: 0x000A21E8, Port5: 0x000A29E8,

Table 1604: F1 Calibration Result Register (n=0–6) 1271
Offset: Port0: 0x000A01EC, Port1: 0x000A09EC, Port2: 0x000A11EC, Port3: 0x000A19EC, Port4: 0x000A21EC, Port5: 0x000A29EC,

Table 1605: RX DCC Calibration Binary Register (n=0–6) 1272
Offset: Port0: 0x000A01F0, Port1: 0x000A09F0, Port2: 0x000A11F0, Port3: 0x000A19F0, Port4: 0x000A21F0, Port5: 0x000A29F0,

Table 1606: TX DCC Calibration Binary Register (n=0–6) 1272
Offset: Port0: 0x000A01F4, Port1: 0x000A09F4, Port2: 0x000A11F4, Port3: 0x000A19F4, Port4: 0x000A21F4, Port5: 0x000A29F4,

Table 1607: Squelch Control 2 Register (n=0–6) 1272
Offset: Port0: 0x000A01F8, Port1: 0x000A09F8, Port2: 0x000A11F8, Port3: 0x000A19F8, Port4: 0x000A21F8, Port5: 0x000A29F8,

Table 1608: RX Control 7 Register (n=0–6) 1272
Offset: Port0: 0x000A0200, Port1: 0x000A0A00, Port2: 0x000A1200, Port3: 0x000A1A00, Port4: 0x000A2200, Port5: 0x000A2A00,

Table 1609: AC JTAG Control 0 Register (n=0–6) 1273
Offset: Port0: 0x000A0204, Port1: 0x000A0A04, Port2: 0x000A1204, Port3: 0x000A1A04, Port4: 0x000A2204, Port5: 0x000A2A04,

Table 1610: DFE F1p5 and F2p5 Control and Other Register (n=0–6) 1273
Offset: Port0: 0x000A0208, Port1: 0x000A0A08, Port2: 0x000A1208, Port3: 0x000A1A08, Port4: 0x000A2208, Port5: 0x000A2A08,

Table 1611: FFE Control Register (n=0–6) 1273
Offset: Port0: 0x000A020C, Port1: 0x000A0A0C, Port2: 0x000A120C, Port3: 0x000A1A0C, Port4: 0x000A220C, Port5: 0x000A2A0C,

Table 1612: DFE 14 Register (n=0–6) 1274
Offset: Port0: 0x000A0210, Port1: 0x000A0A10, Port2: 0x000A1210, Port3: 0x000A1A10, Port4: 0x000A2210, Port5: 0x000A2A10,

Table 1613: Frame Detection Control 0 Register (n=0–6) 1274
Offset: Port0: 0x000A0214, Port1: 0x000A0A14, Port2: 0x000A1214, Port3: 0x000A1A14, Port4: 0x000A2214, Port5: 0x000A2A14,

Table 1614: PLL Calibration Control 11 Register (n=0–6) 1274
Offset: Port0: 0x000A0218, Port1: 0x000A0A18, Port2: 0x000A1218, Port3: 0x000A1A18, Port4: 0x000A2218, Port5: 0x000A2A18,

Table 1615: Frame Detection Control 2 Register (n=0–6) 1275
Offset: Port0: 0x000A021C, Port1: 0x000A0A1C, Port2: 0x000A121C, Port3: 0x000A1A1C, Port4: 0x000A221C, Port5: 0x000A2A1C,

Table 1616: Frame Detection Control 3 Register (n=0–6) 1275
Offset: Port0: 0x000A0220, Port1: 0x000A0A20, Port2: 0x000A1220, Port3: 0x000A1A20, Port4: 0x000A2220, Port5: 0x000A2A20,

Table 1617: TX Training Control 0 Register (n=0–6) 1275
Offset: Port0: 0x000A022C, Port1: 0x000A0A2C, Port2: 0x000A122C, Port3: 0x000A1A2C, Port4: 0x000A222C, Port5: 0x000A2A2C,

Table 1618: TRX Training Control 1 Register (n=0–6) 1276
Offset: Port0: 0x000A0230, Port1: 0x000A0A30, Port2: 0x000A1230, Port3: 0x000A1A30, Port4: 0x000A2230, Port5: 0x000A2A30,

Table 1619: RX Training Control 0 Register (n=0–6) 1277
Offset: Port0: 0x000A0234, Port1: 0x000A0A34, Port2: 0x000A1234, Port3: 0x000A1A34, Port4: 0x000A2234, Port5: 0x000A2A34,

Table 1620: RX Training Control 1 Register (n=0–6) 1277
Offset: Port0: 0x000A0238, Port1: 0x000A0A38, Port2: 0x000A1238, Port3: 0x000A1A38, Port4: 0x000A2238, Port5: 0x000A2A38,

Table 1621: RX Training Control 2 Register (n=0–6) 1277
Offset: Port0: 0x000A023C, Port1: 0x000A0A3C, Port2: 0x000A123C, Port3: 0x000A1A3C, Port4: 0x000A223C, Port5: 0x000A2A3C,

Table 1622: RX Training Control 3 Register (n=0–6) 1277
Offset: Port0: 0x000A0240, Port1: 0x000A0A40, Port2: 0x000A1240, Port3: 0x000A1A40, Port4: 0x000A2240, Port5: 0x000A2A40,

Table 1623: RX Training Control 4 Register (n=0–6) 1278
Offset: Port0: 0x000A0244, Port1: 0x000A0A44, Port2: 0x000A1244, Port3: 0x000A1A44, Port4: 0x000A2244, Port5: 0x000A2A44,

Table 1624: RX Training Control 5 Register (n=0–6) 1278
Offset: Port0: 0x000A0248, Port1: 0x000A0A48, Port2: 0x000A1248, Port3: 0x000A1A48, Port4: 0x000A2248, Port5: 0x000A2A48,

Table 1625: RX Training Control 6 Register (n=0–6) 1279
Offset: Port0: 0x000A024C, Port1: 0x000A0A4C, Port2: 0x000A124C, Port3: 0x000A1A4C, Port4: 0x000A224C, Port5: 0x000A2A4C,

Table 1626: RX Training Control 7 Register (n=0–6) 1279
Offset: Port0: 0x000A0250, Port1: 0x000A0A50, Port2: 0x000A1250, Port3: 0x000A1A50, Port4: 0x000A2250, Port5: 0x000A2A50,

Table 1627: RX Training Control 8 Register (n=0–6) 1279
Offset: Port0: 0x000A0254, Port1: 0x000A0A54, Port2: 0x000A1254, Port3: 0x000A1A54, Port4: 0x000A2254, Port5: 0x000A2A54,

Table 1628: RX Training Control 9 Register (n=0–6) 1280
Offset: Port0: 0x000A0258, Port1: 0x000A0A58, Port2: 0x000A1258, Port3: 0x000A1A58, Port4: 0x000A2258, Port5: 0x000A2A58,

Table 1629: RX Training Control 10 Register (n=0–6) 1280
Offset: Port0: 0x000A025C, Port1: 0x000A0A5C, Port2: 0x000A125C, Port3: 0x000A1A5C, Port4: 0x000A225C, Port5: 0x000A2A5C,

Table 1630: RX Training Control 11 Register (n=0–6) 1280
Offset: Port0: 0x000A0260, Port1: 0x000A0A60, Port2: 0x000A1260, Port3: 0x000A1A60, Port4: 0x000A2260, Port5: 0x000A2A60,

Table 1631: RX Training Control 12 Register (n=0–6) 1281
Offset: Port0: 0x000A0264, Port1: 0x000A0A64, Port2: 0x000A1264, Port3: 0x000A1A64, Port4: 0x000A2264, Port5: 0x000A2A64,

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1632: TX Training Control 0 Register (n=0–6) 1281
Offset: Port0: 0x000A0268, Port1: 0x000A0A68, Port2: 0x000A1268, Port3: 0x000A1A68, Port4: 0x000A2268, Port5: 0x000A2A68,

Table 1633: TX Training Control 1 Register (n=0–6) 1281
Offset: Port0: 0x000A026C, Port1: 0x000A0A6C, Port2: 0x000A126C, Port3: 0x000A1A6C, Port4: 0x000A226C, Port5: 0x000A2A6C,

Table 1634: TRX Training Control 2 Register (n=0–6) 1282
Offset: Port0: 0x000A0270, Port1: 0x000A0A70, Port2: 0x000A1270, Port3: 0x000A1A70, Port4: 0x000A2270, Port5: 0x000A2A70,

Table 1635: TRX Training Control 4 Register (n=0–6) 1282
Offset: Port0: 0x000A0278, Port1: 0x000A0A78, Port2: 0x000A1278, Port3: 0x000A1A78, Port4: 0x000A2278, Port5: 0x000A2A78,

Table 1636: DFE Adaptation Control 0 Register (n=0–6) 1282
Offset: Port0: 0x000A027C, Port1: 0x000A0A7C, Port2: 0x000A127C, Port3: 0x000A1A7C, Port4: 0x000A227C, Port5: 0x000A2A7C,

Table 1637: DFE Adaptation Control 1 Register (n=0–6) 1283
Offset: Port0: 0x000A0280, Port1: 0x000A0A80, Port2: 0x000A1280, Port3: 0x000A1A80, Port4: 0x000A2280, Port5: 0x000A2A80,

Table 1638: DFE Adaptation Control 2 Register (n=0–6) 1283
Offset: Port0: 0x000A0284, Port1: 0x000A0A84, Port2: 0x000A1284, Port3: 0x000A1A84, Port4: 0x000A2284, Port5: 0x000A2A84,

Table 1639: TX Amplitude Setting for PIPE Register (n=0–6) 1283
Offset: Port0: 0x000A0288, Port1: 0x000A0A88, Port2: 0x000A1288, Port3: 0x000A1A88, Port4: 0x000A2288, Port5: 0x000A2A88,

Table 1640: TX Margin Selection Register (n=0–6) 1283
Offset: Port0: 0x000A0290, Port1: 0x000A0A90, Port2: 0x000A1290, Port3: 0x000A1A90, Port4: 0x000A2290, Port5: 0x000A2A90,

Table 1641: Tx_Amp_Adj and Tx_Emph_En Register (n=0–6) 1284
Offset: Port0: 0x000A0294, Port1: 0x000A0A94, Port2: 0x000A1294, Port3: 0x000A1A94, Port4: 0x000A2294, Port5: 0x000A2A94,

Table 1642: TX Amplitude Index Limit Register (n=0–6) 1284
Offset: Port0: 0x000A029C, Port1: 0x000A0A9C, Port2: 0x000A129C, Port3: 0x000A1A9C, Port4: 0x000A229C, Port5: 0x000A2A9C,

Table 1643: TX Emphasis Index Minimum Value Register (n=0–6) 1285
Offset: Port0: 0x000A02A0, Port1: 0x000A0AA0, Port2: 0x000A12A0, Port3: 0x000A1AA0, Port4: 0x000A22A0, Port5: 0x000A2AA0,

Table 1644: TRX Training Control 5 Register (n=0–6) 1285
Offset: Port0: 0x000A02A4, Port1: 0x000A0AA4, Port2: 0x000A12A4, Port3: 0x000A1AA4, Port4: 0x000A22A4, Port5: 0x000A2AA4,

Table 1645: Interrupt 1 Register (n=0–6) 1285
Offset: Port0: 0x000A02AC, Port1: 0x000A0AAC, Port2: 0x000A12AC, Port3: 0x000A1AAC, Port4: 0x000A22AC, Port5: 0x000A2AAC,

Table 1646: Interrupt 3 Register (n=0–6) 1286
Offset: Port0: 0x000A02B4, Port1: 0x000A0AB4, Port2: 0x000A12B4, Port3: 0x000A1AB4, Port4: 0x000A22B4, Port5: 0x000A2AB4,

Table 1647: Interrupt 5 Register (n=0–6) 1287
Offset: Port0: 0x000A02BC, Port1: 0x000A0ABC, Port2: 0x000A12BC, Port3: 0x000A1ABC, Port4: 0x000A22BC, Port5: 0x000A2ABC,

Table 1648: TX Amplitude Default Register (n=0–6) 1287
Offset: Port0: 0x000A02DC, Port1: 0x000A0ADC, Port2: 0x000A12DC, Port3: 0x000A1ADC, Port4: 0x000A22DC, Port5: 0x000A2ADC,

Copyright © 2015 Marvell

Table 1649: Value of Signals to Analog Selection Register (n=0–6) 1287
Offset: Port0: 0x000A02E0, Port1: 0x000A0AE0, Port2: 0x000A12E0, Port3: 0x000A1AE0, Port4: 0x000A22E0, Port5: 0x000A2AE0,

Table 1650: EOM Phase Control Register (n=0–6) 1288
Offset: Port0: 0x000A02EC, Port1: 0x000A0AEC, Port2: 0x000A12EC, Port3: 0x000A1AEC, Port4: 0x000A22EC, Port5: 0x000A2AEC,

Table 1651: EOM Calibration Control 0 Register (n=0–6) 1288
Offset: Port0: 0x000A0308, Port1: 0x000A0B08, Port2: 0x000A1308, Port3: 0x000A1B08, Port4: 0x000A2308, Port5: 0x000A2B08,

Table 1652: EOM Calibration Control 1 Register (n=0–6) 1288
Offset: Port0: 0x000A030C, Port1: 0x000A0B0C, Port2: 0x000A130C, Port3: 0x000A1B0C, Port4: 0x000A230C, Port5: 0x000A2B0C,

Table 1653: TX Train Result Register (n=0–6) 1288
Offset: Port0: 0x000A0310, Port1: 0x000A0B10, Port2: 0x000A1310, Port3: 0x000A1B10, Port4: 0x000A2310, Port5: 0x000A2B10,

Table 1654: TX Train Result Register (n=0–6) 1289
Offset: Port0: 0x000A0314, Port1: 0x000A0B14, Port2: 0x000A1314, Port3: 0x000A1B14, Port4: 0x000A2314, Port5: 0x000A2B14,

Table 1655: Adapted Edge Result Register (n=0–6) 1289
Offset: Port0: 0x000A0318, Port1: 0x000A0B18, Port2: 0x000A1318, Port3: 0x000A1B18, Port4: 0x000A2318, Port5: 0x000A2B18,

Table 1656: TX Train Preset Register (n=0–6) 1290
Offset: Port0: 0x000A031C, Port1: 0x000A0B1C, Port2: 0x000A131C, Port3: 0x000A1B1C, Port4: 0x000A231C, Port5: 0x000A2B1C,

Table 1657: Phase Offset Range Register (n=0–6) 1290
Offset: Port0: 0x000A0320, Port1: 0x000A0B20, Port2: 0x000A1320, Port3: 0x000A1B20, Port4: 0x000A2320, Port5: 0x000A2B20,

Table 1658: Saved DFE Values 0 Register (n=0–6) 1290
Offset: Port0: 0x000A0324, Port1: 0x000A0B24, Port2: 0x000A1324, Port3: 0x000A1B24, Port4: 0x000A2324, Port5: 0x000A2B24,

Table 1659: Saved DFE Values 1 Register (n=0–6) 1291
Offset: Port0: 0x000A0328, Port1: 0x000A0B28, Port2: 0x000A1328, Port3: 0x000A1B28, Port4: 0x000A2328, Port5: 0x000A2B28,

Table 1660: Saved DFE Values 2 Register (n=0–6) 1291
Offset: Port0: 0x000A032C, Port1: 0x000A0B2C, Port2: 0x000A132C, Port3: 0x000A1B2C, Port4: 0x000A232C, Port5: 0x000A2B2C,

Table 1661: RX Train Default Values Register (n=0–6) 1291
Offset: Port0: 0x000A0330, Port1: 0x000A0B30, Port2: 0x000A1330, Port3: 0x000A1B30, Port4: 0x000A2330, Port5: 0x000A2B30,

Table 1662: Status Detection Timeout Control Register (n=0–6) 1291
Offset: Port0: 0x000A0334, Port1: 0x000A0B34, Port2: 0x000A1334, Port3: 0x000A1B34, Port4: 0x000A2334, Port5: 0x000A2B34,

Table 1663: TX Amplitude Default Register (n=0–6) 1292
Offset: Port0: 0x000A0338, Port1: 0x000A0B38, Port2: 0x000A1338, Port3: 0x000A1B38, Port4: 0x000A2338, Port5: 0x000A2B38,

Table 1664: TX Amplitude Default Register (n=0–6) 1292
Offset: Port0: 0x000A033C, Port1: 0x000A0B3C, Port2: 0x000A133C, Port3: 0x000A1B3C, Port4: 0x000A233C, Port5: 0x000A2B3C,

Table 1665: DFE Adaptation Control 3 Register (n=0–6) 1292
Offset: Port0: 0x000A0340, Port1: 0x000A0B40, Port2: 0x000A1340, Port3: 0x000A1B40, Port4: 0x000A2340, Port5: 0x000A2B40,

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1666: RX Training Control 14 Register (n=0–6) 1293
Offset: Port0: 0x000A0348, Port1: 0x000A0B48, Port2: 0x000A1348, Port3: 0x000A1B48, Port4: 0x000A2348, Port5: 0x000A2B48,

Table 1667: RX Training Control 15 Register (n=0–6) 1293
Offset: Port0: 0x000A034C, Port1: 0x000A0B4C, Port2: 0x000A134C, Port3: 0x000A1B4C, Port4: 0x000A234C, Port5: 0x000A2B4C,

Table 1668: RX Training Control 16 Register (n=0–6) 1293
Offset: Port0: 0x000A0350, Port1: 0x000A0B50, Port2: 0x000A1350, Port3: 0x000A1B50, Port4: 0x000A2350, Port5: 0x000A2B50,

Table 1669: TX Training Control 6 Register (n=0–6) 1293
Offset: Port0: 0x000A035C, Port1: 0x000A0B5C, Port2: 0x000A135C, Port3: 0x000A1B5C, Port4: 0x000A235C, Port5: 0x000A2B5C,

Table 1670: TX Training Control 7 Register (n=0–6) 1293
Offset: Port0: 0x000A0360, Port1: 0x000A0B60, Port2: 0x000A1360, Port3: 0x000A1B60, Port4: 0x000A2360, Port5: 0x000A2B60,

Table 1671: TX Training Control 7 Register (n=0–6) 1294
Offset: Port0: 0x000A0364, Port1: 0x000A0B64, Port2: 0x000A1364, Port3: 0x000A1B64, Port4: 0x000A2364, Port5: 0x000A2B64,

Table 1672: TRX Training Control 6 Register (n=0–6) 1294
Offset: Port0: 0x000A036C, Port1: 0x000A0B6C, Port2: 0x000A136C, Port3: 0x000A1B6C, Port4: 0x000A236C, Port5: 0x000A2B6C,

Table 1673: Calibration Control 16 Register (n=0–6) 1294
Offset: Port0: 0x000A0370, Port1: 0x000A0B70, Port2: 0x000A1370, Port3: 0x000A1B70, Port4: 0x000A2370, Port5: 0x000A2B70,

Table 1674: VDD Calibration Control 4 Register (n=0–6) 1295
Offset: Port0: 0x000A0374, Port1: 0x000A0B74, Port2: 0x000A1374, Port3: 0x000A1B74, Port4: 0x000A2374, Port5: 0x000A2B74,

Table 1675: F1 Buffer Offset Calibration Control 0 Register (n=0–6) 1295
Offset: Port0: 0x000A0378, Port1: 0x000A0B78, Port2: 0x000A1378, Port3: 0x000A1B78, Port4: 0x000A2378, Port5: 0x000A2B78,

Table 1676: TX Default 4 Register (n=0–6) 1295
Offset: Port0: 0x000A037C, Port1: 0x000A0B7C, Port2: 0x000A137C, Port3: 0x000A1B7C, Port4: 0x000A237C, Port5: 0x000A2B7C,

Table 1677: PLL Calibration Control 2 Register (n=0–6) 1295
Offset: Port0: 0x000A0380, Port1: 0x000A0B80, Port2: 0x000A1380, Port3: 0x000A1B80, Port4: 0x000A2380, Port5: 0x000A2B80,

Table 1678: PLL Calibration Control 3 Register (n=0–6) 1296
Offset: Port0: 0x000A0384, Port1: 0x000A0B84, Port2: 0x000A1384, Port3: 0x000A1B84, Port4: 0x000A2384, Port5: 0x000A2B84,

Table 1679: PLL Calibration Control 4 Register (n=0–6) 1296
Offset: Port0: 0x000A0388, Port1: 0x000A0B88, Port2: 0x000A1388, Port3: 0x000A1B88, Port4: 0x000A2388, Port5: 0x000A2B88,

Table 1680: PLL Calibration Control 5 Register (n=0–6) 1296
Offset: Port0: 0x000A038C, Port1: 0x000A0B8C, Port2: 0x000A138C, Port3: 0x000A1B8C, Port4: 0x000A238C, Port5: 0x000A2B8C,

Table 1681: PLL Calibration Control 6 Register (n=0–6) 1296
Offset: Port0: 0x000A0390, Port1: 0x000A0B90, Port2: 0x000A1390, Port3: 0x000A1B90, Port4: 0x000A2390, Port5: 0x000A2B90,

Table 1682: PLL Calibration Control 8 Register (n=0–6) 1297
Offset: Port0: 0x000A0398, Port1: 0x000A0B98, Port2: 0x000A1398, Port3: 0x000A1B98, Port4: 0x000A2398, Port5: 0x000A2B98,

Copyright © 2015 Marvell

Table 1683: FFE Control 2 Register (n=0–6) 1297
Offset: Port0: 0x000A039C, Port1: 0x000A0B9C, Port2: 0x000A139C, Port3: 0x000A1B9C, Port4: 0x000A239C, Port5: 0x000A2B9C,

Table 1684: Miscellaneous Control 3 Register (n=0–6) 1297
Offset: Port0: 0x000A03A0, Port1: 0x000A0BA0, Port2: 0x000A13A0, Port3: 0x000A1BA0, Port4: 0x000A23A0, Port5: 0x000A2BA0,

Table 1685: Miscellaneous Control 4 Register (n=0–6) 1298
Offset: Port0: 0x000A03A4, Port1: 0x000A0BA4, Port2: 0x000A13A4, Port3: 0x000A1BA4, Port4: 0x000A23A4, Port5: 0x000A2BA4,

Table 1686: TRX Training Control 7 Register (n=0–6) 1298
Offset: Port0: 0x000A03A8, Port1: 0x000A0BA8, Port2: 0x000A13A8, Port3: 0x000A1BA8, Port4: 0x000A23A8, Port5: 0x000A2BA8,

Table 1687: TRX Training Control 8 Register (n=0–6) 1298
Offset: Port0: 0x000A03AC, Port1: 0x000A0BAC, Port2: 0x000A13AC, Port3: 0x000A1BAC, Port4: 0x000A23AC, Port5: 0x000A2BAC,

Table 1688: DFE Step Size 2 Register (n=0–6) 1298
Offset: Port0: 0x000A03B0, Port1: 0x000A0BB0, Port2: 0x000A13B0, Port3: 0x000A1BB0, Port4: 0x000A23B0, Port5: 0x000A2BB0,

Table 1689: TRX Training Control 10 Register (n=0–6) 1300
Offset: Port0: 0x000A03B4, Port1: 0x000A0BB4, Port2: 0x000A13B4, Port3: 0x000A1BB4, Port4: 0x000A23B4, Port5: 0x000A2BB4,

Table 1690: PLL Calibration Control Register (n=0–6) 1300
Offset: Port0: 0x000A03B8, Port1: 0x000A0BB8, Port2: 0x000A13B8, Port3: 0x000A1BB8, Port4: 0x000A23B8, Port5: 0x000A2BB8,

Table 1691: Align 90 Calibration Control 0 Register (n=0–6) 1301
Offset: Port0: 0x000A03BC, Port1: 0x000A0BBC, Port2: 0x000A13BC, Port3: 0x000A1BBC, Port4: 0x000A23BC, Port5: 0x000A2BBC,

Table 1692: Digital Test Bus Configuration Register (n=0–6) 1301
Offset: Port0: 0x000A03C0, Port1: 0x000A0BC0, Port2: 0x000A13C0, Port3: 0x000A1BC0, Port4: 0x000A23C0, Port5: 0x000A2BC0,

Table 1693: Digital Test Bus Register (n=0–6) 1301
Offset: Port0: 0x000A03C4, Port1: 0x000A0BC4, Port2: 0x000A13C4, Port3: 0x000A1BC4, Port4: 0x000A23C4, Port5: 0x000A2BC4,

Table 1694: Analog Control 1 Register (n=0–6) 1301
Offset: Port0: 0x000A03C8, Port1: 0x000A0BC8, Port2: 0x000A13C8, Port3: 0x000A1BC8, Port4: 0x000A23C8, Port5: 0x000A2BC8,

Table 1695: Analog Control 2 Register (n=0–6) 1302
Offset: Port0: 0x000A03CC, Port1: 0x000A0BCC, Port2: 0x000A13CC, Port3: 0x000A1BCC, Port4: 0x000A23CC, Port5: 0x000A2BCC,

Table 1696: Analog Control 3 Register (n=0–6) 1302
Offset: Port0: 0x000A03D0, Port1: 0x000A0BD0, Port2: 0x000A13D0, Port3: 0x000A1BD0, Port4: 0x000A23D0, Port5: 0x000A2BD0,

Table 1697: Analog Control 4 Register (n=0–6) 1302
Offset: Port0: 0x000A03D4, Port1: 0x000A0BD4, Port2: 0x000A13D4, Port3: 0x000A1BD4, Port4: 0x000A23D4, Port5: 0x000A2BD4,

Table 1698: TX Training Control 25 Register (n=0–6) 1303
Offset: Port0: 0x000A03D8, Port1: 0x000A0BD8, Port2: 0x000A13D8, Port3: 0x000A1BD8, Port4: 0x000A23D8, Port5: 0x000A2BD8,

Table 1699: TX Default 5 Register (n=0–6) 1303
Offset: Port0: 0x000A03DC, Port1: 0x000A0BDC, Port2: 0x000A13DC, Port3: 0x000A1BDC, Port4: 0x000A23DC, Port5: 0x000A2BDC,

Copyright © 2015 Marvell

Table 1700: Clk150m Control 3 Register (n=0–6) 1303
Offset: Port0: 0x000A03E0, Port1: 0x000A0BE0, Port2: 0x000A13E0, Port3: 0x000A1BE0, Port4: 0x000A23E0, Port5: 0x000A2BE0,

Table 1701: Signal Detection Calibration Register (n=0–6) 1303
Offset: Port0: 0x000A03E4, Port1: 0x000A0BE4, Port2: 0x000A13E4, Port3: 0x000A1BE4, Port4: 0x000A23E4, Port5: 0x000A2BE4,

Table 1702: Analog Control 5 Register (n=0–6) 1304
Offset: Port0: 0x000A03E8, Port1: 0x000A0BE8, Port2: 0x000A13E8, Port3: 0x000A1BE8, Port4: 0x000A23E8, Port5: 0x000A2BE8,

Table 1703: TX Default 7 Register (n=0–6) 1304
Offset: Port0: 0x000A03EC, Port1: 0x000A0BEC, Port2: 0x000A13EC, Port3: 0x000A1BEC, Port4: 0x000A23EC, Port5: 0x000A2BEC,

Table 1704: TRX Training Control 9 Register (n=0–6) 1304
Offset: Port0: 0x000A03F0, Port1: 0x000A0BF0, Port2: 0x000A13F0, Port3: 0x000A1BF0, Port4: 0x000A23F0, Port5: 0x000A2BF0,

Table 1705: TX Default 6 Register (n=0–6) 1305
Offset: Port0: 0x000A03F4, Port1: 0x000A0BF4, Port2: 0x000A13F4, Port3: 0x000A1BF4, Port4: 0x000A23F4, Port5: 0x000A2BF4,

Table 1706: Squelch OOB Register (n=0–6) 1305
Offset: Port0: 0x000A03F8, Port1: 0x000A0BF8, Port2: 0x000A13F8, Port3: 0x000A1BF8, Port4: 0x000A23F8, Port5: 0x000A2BF8,

Table 1707: Squelch Detect Calibration and VDD Calibration Result Register (n=0–6) 1305
Offset: Port0: 0x000A0400, Port1: 0x000A0C00, Port2: 0x000A1400, Port3: 0x000A1C00, Port4: 0x000A2400, Port5: 0x000A2C00,

Table 1708: Align 90 Calibration Result Register (n=0–6) 1306
Offset: Port0: 0x000A0404, Port1: 0x000A0C04, Port2: 0x000A1404, Port3: 0x000A1C04, Port4: 0x000A2404, Port5: 0x000A2C04,

Table 1709: DCC Calibration Result Register (n=0–6) 1306
Offset: Port0: 0x000A0408, Port1: 0x000A0C08, Port2: 0x000A1408, Port3: 0x000A1C08, Port4: 0x000A2408, Port5: 0x000A2C08,

Table 1710: Calibration Result Register (n=0–6) 1306
Offset: Port0: 0x000A040C, Port1: 0x000A0C0C, Port2: 0x000A140C, Port3: 0x000A1C0C, Port4: 0x000A240C, Port5: 0x000A2C0C,

Table 1711: RX DCC Calibration Result Register (n=0–6) 1306
Offset: Port0: 0x000A0410, Port1: 0x000A0C10, Port2: 0x000A1410, Port3: 0x000A1C10, Port4: 0x000A2410, Port5: 0x000A2C10,

Table 1712: Sampler Calibration Result Register (n=0–6) 1307
Offset: Port0: 0x000A0414, Port1: 0x000A0C14, Port2: 0x000A1414, Port3: 0x000A1C14, Port4: 0x000A2414, Port5: 0x000A2C14,

Table 1713: CDR Control Register (n=0–6) 1307
Offset: Port0: 0x000A0418, Port1: 0x000A0C18, Port2: 0x000A1418, Port3: 0x000A1C18, Port4: 0x000A2418, Port5: 0x000A2C18,

Table 1714: CDR Control Register (n=0–6) 1308
Offset: Port0: 0x000A041C, Port1: 0x000A0C1C, Port2: 0x000A141C, Port3: 0x000A1C1C, Port4: 0x000A241C, Port5: 0x000A2C1C,

Table 1715: CDR Control Register (n=0–6) 1308
Offset: Port0: 0x000A0420, Port1: 0x000A0C20, Port2: 0x000A1420, Port3: 0x000A1C20, Port4: 0x000A2420, Port5: 0x000A2C20,

Table 1716: CDR Control Register (n=0–6) 1309
Offset: Port0: 0x000A0424, Port1: 0x000A0C24, Port2: 0x000A1424, Port3: 0x000A1C24, Port4: 0x000A2424, Port5: 0x000A2C24,

Table 1717: RX Train Midpoint Register (n=0–6) 1309
Offset: Port0: 0x000A0428, Port1: 0x000A0C28, Port2: 0x000A1428, Port3: 0x000A1C28, Port4: 0x000A2428, Port5: 0x000A2C28,

Table 1718: RX Train Midpoint Register (n=0–6) 1309
Offset: Port0: 0x000A042C, Port1: 0x000A0C2C, Port2: 0x000A142C, Port3: 0x000A1C2C, Port4: 0x000A242C, Port5: 0x000A2C2C,

Table 1719: TX Training Control 9 Register (n=0–6) 1309
Offset: Port0: 0x000A0430, Port1: 0x000A0C30, Port2: 0x000A1430, Port3: 0x000A1C30, Port4: 0x000A2430, Port5: 0x000A2C30,

Table 1720: TX Training Control 10 Register (n=0–6) 1310
Offset: Port0: 0x000A0434, Port1: 0x000A0C34, Port2: 0x000A1434, Port3: 0x000A1C34, Port4: 0x000A2434, Port5: 0x000A2C34,

Table 1721: TX Training Control 11 Register (n=0–6) 1310
Offset: Port0: 0x000A0438, Port1: 0x000A0C38, Port2: 0x000A1438, Port3: 0x000A1C38, Port4: 0x000A2438, Port5: 0x000A2C38,

Table 1722: TX Training Control 12 Register (n=0–6) 1310
Offset: Port0: 0x000A043C, Port1: 0x000A0C3C, Port2: 0x000A143C, Port3: 0x000A1C3C, Port4: 0x000A243C, Port5: 0x000A2C3C,

Table 1723: G1 Setting 3 Register (n=0–6) 1310
Offset: Port0: 0x000A0440, Port1: 0x000A0C40, Port2: 0x000A1440, Port3: 0x000A1C40, Port4: 0x000A2440, Port5: 0x000A2C40,

Table 1724: G1 Setting 4 Register (n=0–6) 1311
Offset: Port0: 0x000A0444, Port1: 0x000A0C44, Port2: 0x000A1444, Port3: 0x000A1C44, Port4: 0x000A2444, Port5: 0x000A2C44,

Table 1725: G2 Setting 3 Register (n=0–6) 1312
Offset: Port0: 0x000A0448, Port1: 0x000A0C48, Port2: 0x000A1448, Port3: 0x000A1C48, Port4: 0x000A2448, Port5: 0x000A2C48,

Table 1726: G2 Setting 4 Register (n=0–6) 1312
Offset: Port0: 0x000A044C, Port1: 0x000A0C4C, Port2: 0x000A144C, Port3: 0x000A1C4C, Port4: 0x000A244C, Port5: 0x000A2C4C,

Table 1727: G3 Setting 3 Register (n=0–6) 1313
Offset: Port0: 0x000A0450, Port1: 0x000A0C50, Port2: 0x000A1450, Port3: 0x000A1C50, Port4: 0x000A2450, Port5: 0x000A2C50,

Table 1728: G3 Setting 4 Register (n=0–6) 1314
Offset: Port0: 0x000A0454, Port1: 0x000A0C54, Port2: 0x000A1454, Port3: 0x000A1C54, Port4: 0x000A2454, Port5: 0x000A2C54,

Table 1729: G4_setting_3 Register (n=0–6) 1314
Offset: Port0: 0x000A0458, Port1: 0x000A0C58, Port2: 0x000A1458, Port3: 0x000A1C58, Port4: 0x000A2458, Port5: 0x000A2C58,

Table 1730: G4 Setting 4 Register (n=0–6) 1315
Offset: Port0: 0x000A045C, Port1: 0x000A0C5C, Port2: 0x000A145C, Port3: 0x000A1C5C, Port4: 0x000A245C, Port5: 0x000A2C5C,

Table 1731: PCIe TX Preset Table 0 Register (n=0–6) 1315
Offset: Port0: 0x000A0460, Port1: 0x000A0C60, Port2: 0x000A1460, Port3: 0x000A1C60, Port4: 0x000A2460, Port5: 0x000A2C60,

Table 1732: PCIe TX Preset Table 1 Register (n=0–6) 1316
Offset: Port0: 0x000A0464, Port1: 0x000A0C64, Port2: 0x000A1464, Port3: 0x000A1C64, Port4: 0x000A2464, Port5: 0x000A2C64,

Table 1733: PCIe TX Preset Index Register (n=0–6) 1316
Offset: Port0: 0x000A0468, Port1: 0x000A0C68, Port2: 0x000A1468, Port3: 0x000A1C68, Port4: 0x000A2468, Port5: 0x000A2C68,

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1734: TX Power Protection Control Register (n=0–6) 1316
Offset: Port0: 0x000A046C, Port1: 0x000A0C6C, Port2: 0x000A146C, Port3: 0x000A1C6C, Port4: 0x000A246C, Port5: 0x000A2C6C,

Table 1735: DFE Control 27 Register (n=0–6) 1317
Offset: Port0: 0x000A0470, Port1: 0x000A0C70, Port2: 0x000A1470, Port3: 0x000A1C70, Port4: 0x000A2470, Port5: 0x000A2C70,

Table 1736: RX Training for Over Boost Register (n=0–6) 1317
Offset: Port0: 0x000A0474, Port1: 0x000A0C74, Port2: 0x000A1474, Port3: 0x000A1C74, Port4: 0x000A2474, Port5: 0x000A2C74,

Table 1737: TX Training for Over Boost Register (n=0–6) 1318
Offset: Port0: 0x000A0478, Port1: 0x000A0C78, Port2: 0x000A1478, Port3: 0x000A1C78, Port4: 0x000A2478, Port5: 0x000A2C78,

Table 1738: RX Training Mode 0 Register (n=0–6) 1318
Offset: Port0: 0x000A047C, Port1: 0x000A0C7C, Port2: 0x000A147C, Port3: 0x000A1C7C, Port4: 0x000A247C, Port5: 0x000A2C7C,

Table 1739: PCIe RX Preset Hint 1 Register (n=0–6) 1318
Offset: Port0: 0x000A0480, Port1: 0x000A0C80, Port2: 0x000A1480, Port3: 0x000A1C80, Port4: 0x000A2480, Port5: 0x000A2C80,

Table 1740: PCIe RX Preset Hint 2 Register (n=0–6) 1318
Offset: Port0: 0x000A0484, Port1: 0x000A0C84, Port2: 0x000A1484, Port3: 0x000A1C84, Port4: 0x000A2484, Port5: 0x000A2C84,

Table 1741: PCIe RX Preset Hint 3 Register (n=0–6) 1319
Offset: Port0: 0x000A0488, Port1: 0x000A0C88, Port2: 0x000A1488, Port3: 0x000A1C88, Port4: 0x000A2488, Port5: 0x000A2C88,

Table 1742: PCIe RX Preset Hint 4 Register (n=0–6) 1319
Offset: Port0: 0x000A048C, Port1: 0x000A0C8C, Port2: 0x000A148C, Port3: 0x000A1C8C, Port4: 0x000A248C, Port5: 0x000A2C8C,

Table 1743: PCIe RX Preset Hint 5 Register (n=0–6) 1320
Offset: Port0: 0x000A0490, Port1: 0x000A0C90, Port2: 0x000A1490, Port3: 0x000A1C90, Port4: 0x000A2490, Port5: 0x000A2C90,

Table 1744: PCIe RX Preset Hint 6 Register (n=0–6) 1320
Offset: Port0: 0x000A0494, Port1: 0x000A0C94, Port2: 0x000A1494, Port3: 0x000A1C94, Port4: 0x000A2494, Port5: 0x000A2C94,

Table 1745: PCIe RX Preset Hint 7 Register (n=0–6) 1320
Offset: Port0: 0x000A0498, Port1: 0x000A0C98, Port2: 0x000A1498, Port3: 0x000A1C98, Port4: 0x000A2498, Port5: 0x000A2C98,

Table 1746: DFE Control 28 Register (n=0–6) 1320
Offset: Port0: 0x000A049C, Port1: 0x000A0C9C, Port2: 0x000A149C, Port3: 0x000A1C9C, Port4: 0x000A249C, Port5: 0x000A2C9C,

Table 1747: F1_BUF_OS Adjust 0 Register (n=0–6) 1321
Offset: Port0: 0x000A04A0, Port1: 0x000A0CA0, Port2: 0x000A14A0, Port3: 0x000A1CA0, Port4: 0x000A24A0, Port5: 0x000A2CA0,

Table 1748: F1_BUF_OS Adjust 1 Register (n=0–6) 1321
Offset: Port0: 0x000A04A4, Port1: 0x000A0CA4, Port2: 0x000A14A4, Port3: 0x000A1CA4, Port4: 0x000A24A4, Port5: 0x000A2CA4,

Table 1749: F1_BUF_OS Adjust 2 Register (n=0–6) 1322
Offset: Port0: 0x000A04A8, Port1: 0x000A0CA8, Port2: 0x000A14A8, Port3: 0x000A1CA8, Port4: 0x000A24A8, Port5: 0x000A2CA8,

Table 1750: F1_BUF_OS Adjust 3 Register (n=0–6) 1322
Offset: Port0: 0x000A04AC, Port1: 0x000A0CAC, Port2: 0x000A14AC, Port3: 0x000A1CAC, Port4: 0x000A24AC, Port5: 0x000A2CAC,

Copyright © 2015 Marvell

Table 1751: TX Default 8 Register (n=0–6) 1322
Offset: Port0: 0x000A04B0, Port1: 0x000A0CB0, Port2: 0x000A14B0, Port3: 0x000A1CB0, Port4: 0x000A24B0, Port5: 0x000A2CB0,
Table 1752: TX Default 9 Register (n=0–6) 1323
Offset: Port0: 0x000A04B4, Port1: 0x000A0CB4, Port2: 0x000A14B4, Port3: 0x000A1CB4, Port4: 0x000A24B4, Port5: 0x000A2CB4,
Table 1753: TX Default 10 Register (n=0–6) 1323
Offset: Port0: 0x000A04B8, Port1: 0x000A0CB8, Port2: 0x000A14B8, Port3: 0x000A1CB8, Port4: 0x000A24B8, Port5: 0x000A2CB8,
Table 1754: TX Default 11 Register (n=0–6) 1323
Offset: Port0: 0x000A04BC, Port1: 0x000A0CBC, Port2: 0x000A14BC, Port3: 0x000A1CBC, Port4: 0x000A24BC, Port5: 0x000A2CBC,
Table 1755: DFE 30 Register (n=0–6) 1323
Offset: Port0: 0x000A04C0, Port1: 0x000A0CC0, Port2: 0x000A14C0, Port3: 0x000A1CC0, Port4: 0x000A24C0, Port5: 0x000A2CC0,
Table 1756: Internal Digital Reserved 0 Register (n=0–6) 1324
Offset: Port0: 0x000A05EC, Port1: 0x000A0DEC, Port2: 0x000A15EC, Port3: 0x000A1DEC, Port4: 0x000A25EC, Port5: 0x000A2DEC,
Table 1757: Analog Reserved 0 Register (n=0–6) 1324
Offset: Port0: 0x000A05F0, Port1: 0x000A0DF0, Port2: 0x000A15F0, Port3: 0x000A1DF0, Port4: 0x000A25F0, Port5: 0x000A2DF0,
Table 1758: Internal CID 0 Register (n=0–6) 1324
Offset: Port0: 0x000A05F4, Port1: 0x000A0DF4, Port2: 0x000A15F4, Port3: 0x000A1DF4, Port4: 0x000A25F4, Port5: 0x000A2DF4,
Table 1759: Internal CID 1 Register (n=0–6) 1324
Offset: Port0: 0x000A05F8, Port1: 0x000A0DF8, Port2: 0x000A15F8, Port3: 0x000A1DF8, Port4: 0x000A25F8, Port5: 0x000A2DF8,
Table 1760: Lane Configuration 0 Register (n=0–1) 1324
Offset: Port0: 0x000A0600, Port1: 0x000A0E00
Table 1761: Lane Configuration 1 Register (n=0–1) 1326
Offset: Port0: 0x000A0604, Port1: 0x000A0E04
Table 1762: Lane Status 0 Register (n=0–1) 1327
Offset: Port0: 0x000A0608, Port1: 0x000A0E08
Table 1763: Lane Status 1 Register (n=0–1) 1329
Offset: Port0: 0x000A060C, Port1: 0x000A0E0C
Table 1764: Lane Status 2 Register (n=0–1) 1330
Offset: Port0: 0x000A0610, Port1: 0x000A0E10
Table 1765: Lane Beacon Control Register (n=0–1) 1331
Offset: Port0: 0x000A0614, Port1: 0x000A0E14
Table 1766: Lane Configuration 2 Register (n=0–1) 1332
Offset: Port0: 0x000A0618, Port1: 0x000A0E18
Table 1767: Lane Configuration 3 Register (n=0–1) 1333
Offset: Port0: 0x000A061C, Port1: 0x000A0E1C
Table 1768: Lane Configuration 4 Register (n=0–1) 1333
Offset: Port0: 0x000A0620, Port1: 0x000A0E20
Table 1769: Lane Configuration 5 Register (n=0–1) 1335
Offset: Port0: 0x000A0624, Port1: 0x000A0E24
Table 1770: Lane Status 3 Register (n=0–1) 1335
Offset: Port0: 0x000A0628, Port1: 0x000A0E28

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1771: Lane TX Common Mode Configuration Register (n=0–1) 1336
Offset: Port0: 0x000A062C, Port1: 0x000A0E2C
Table 1772: USB Datapath Configuration 1 Register (n=0–1) 1336
Offset: Port0: 0x000A0684, Port1: 0x000A0E84
Table 1773: Lane Equalization Configuration 0 Register (n=0–1) 1337
Offset: Port0: 0x000A069C, Port1: 0x000A0E9C
Table 1774: Lane Equalization Configuration 1 Register (n=0–1) 1337
Offset: Port0: 0x000A06A0, Port1: 0x000A0EA0
Table 1775: Lane Equalization Configuration 2 Register (n=0–1) 1337
Offset: Port0: 0x000A06A4, Port1: 0x000A0EA4
Table 1776: Lane Equalization Preset Configuration 0 Register (n=0–1) 1337
Offset: Port0: 0x000A06A8, Port1: 0x000A0EA8
Table 1777: Lane Equalization Preset Configuration 1 Register (n=0–1) 1338
Offset: Port0: 0x000A06AC, Port1: 0x000A0EAC
Table 1778: Lane Equalization Preset Configuration 2 Register (n=0–1) 1338
Offset: Port0: 0x000A06B0, Port1: 0x000A0EB0
Table 1779: Lane Equalization Preset Configuration 3 Register (n=0–1) 1338
Offset: Port0: 0x000A06B4, Port1: 0x000A0EB4
Table 1780: Lane Equalization Preset Configuration 4 Register (n=0–1) 1339
Offset: Port0: 0x000A06B8, Port1: 0x000A0EB8
Table 1781: Lane Equalization Preset Configuration 5 Register (n=0–1) 1339
Offset: Port0: 0x000A06BC, Port1: 0x000A0EBC
Table 1782: Lane Equalization Preset Configuration 6 Register (n=0–1) 1339
Offset: Port0: 0x000A06C0, Port1: 0x000A0EC0
Table 1783: Lane Equalization Preset Configuration 7 Register (n=0–1) 1339
Offset: Port0: 0x000A06C4, Port1: 0x000A0EC4
Table 1784: Lane Equalization Preset Configuration 8 Register (n=0–1) 1340
Offset: Port0: 0x000A06C8, Port1: 0x000A0EC8
Table 1785: Lane Equalization Preset Configuration 9 Register (n=0–1) 1340
Offset: Port0: 0x000A06CC, Port1: 0x000A0ECC
Table 1786: Lane Equalization Preset Configuration 10 Register (n=0–1) 1340
Offset: Port0: 0x000A06D0, Port1: 0x000A0ED0
Table 1787: Lane Equalization Preset Configuration 11 Register (n=0–1) 1340
Offset: Port0: 0x000A06D4, Port1: 0x000A0ED4
Table 1788: Lane Equalization Preset Configuration 12 Register (n=0–1) 1341
Offset: Port0: 0x000A06D8, Port1: 0x000A0ED8
Table 1789: Lane Equalization Preset Configuration 13 Register (n=0–1) 1341
Offset: Port0: 0x000A06DC, Port1: 0x000A0EDC
Table 1790: Lane Equalization Preset Configuration 14 Register (n=0–1) 1341
Offset: Port0: 0x000A06E0, Port1: 0x000A0EE0
Table 1791: Lane Equalization Preset Configuration 15 Register (n=0–1) 1341
Offset: Port0: 0x000A06E4, Port1: 0x000A0EE4
Table 1792: Lane Equalization Preset Configuration 16 Register (n=0–1) 1342
Offset: Port0: 0x000A06E8, Port1: 0x000A0EE8
Table 1793: Lane Equalization Preset Configuration 17 Register (n=0–1) 1342
Offset: Port0: 0x000A06EC, Port1: 0x000A0EEC
Table 1794: Lane Equalization Coefficient Max Setting 0 Register (n=0–1) 1342
Offset: Port0: 0x000A06F0, Port1: 0x000A0EF0

Copyright © 2015 Marvell

Table 1795: Lane Equalization Coefficient Max Setting 1 Register (n=0–1) 1342
Offset: Port0: 0x000A06F4, Port1: 0x000A0EF4
Table 1796: Lane Equalization Remote Setting Register (n=0–1) 1343
Offset: Port0: 0x000A06F8, Port1: 0x000A0EF8
Table 1797: Lane Equalization Status Register (n=0–1) 1343
Offset: Port0: 0x000A06FC, Port1: 0x000A0EFC
Table 1798: PIPE Revision Register (n=0–1) 1343
Offset: Port0: 0x000A0700, Port1: 0x000A0F00
Table 1799: Reset and Clock Control Register (n=0–1) 1343
Offset: Port0: 0x000A0704, Port1: 0x000A0F04
Table 1800: Test Mode Control Register (n=0–1) 1344
Offset: Port0: 0x000A0708, Port1: 0x000A0F08
Table 1801: Clock Source Low Register (n=0–1) 1346
Offset: Port0: 0x000A070C, Port1: 0x000A0F0C
Table 1802: Clock Source High Register (n=0–1) 1347
Offset: Port0: 0x000A0710, Port1: 0x000A0F10
Table 1803: Trigger Register (n=0–1) 1348
Offset: Port0: 0x000A0714, Port1: 0x000A0F14
Table 1804: Miscellaneous Control Register (n=0–1) 1349
Offset: Port0: 0x000A0718, Port1: 0x000A0F18
Table 1805: CLKREQ# Timing Control Register (n=0–1) 1351
Offset: Port0: 0x000A071C, Port1: 0x000A0F1C
Table 1806: Datapath Configuration Register (n=0–1) 1352
Offset: Port0: 0x000A0720, Port1: 0x000A0F20
Table 1807: Symbol Alignment Good Symbol Weight Register (n=0–1) 1353
Offset: Port0: 0x000A0724, Port1: 0x000A0F24
Table 1808: Symbol Alignment Aligned COM Weight Register (n=0–1) 1353
Offset: Port0: 0x000A0728, Port1: 0x000A0F28
Table 1809: Symbol Alignment Bad Symbol Weight Register (n=0–1) 1353
Offset: Port0: 0x000A072C, Port1: 0x000A0F2C
Table 1810: Symbol Alignment Misaligned COM Weight Register (n=0–1) 1354
Offset: Port0: 0x000A0730, Port1: 0x000A0F30
Table 1811: Symbol Alignment Squelch Detect Register (n=0–1) 1354
Offset: Port0: 0x000A0734, Port1: 0x000A0F34
Table 1812: Symbol Alignment Comma Pattern Register (n=0–1) 1354
Offset: Port0: 0x000A0738, Port1: 0x000A0F38
Table 1813: Power Management Timing Parameter 1 Register (n=0–1) 1355
Offset: Port0: 0x000A0740, Port1: 0x000A0F40
Table 1814: Power Management Timing Parameter 2 Register (n=0–1) 1355
Offset: Port0: 0x000A0744, Port1: 0x000A0F44
Table 1815: Counter Lane and Type Register (n=0–1) 1355
Offset: Port0: 0x000A0748, Port1: 0x000A0F48
Table 1816: Counter Low Register (n=0–1) 1356
Offset: Port0: 0x000A074C, Port1: 0x000A0F4C
Table 1817: Counter High Register (n=0–1) 1356
Offset: Port0: 0x000A0750, Port1: 0x000A0F50
Table 1818: PM Serdes Control Override Register (n=0–1) 1357
Offset: Port0: 0x000A0754, Port1: 0x000A0F54

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1819: PM Datapath and Clock Control Override Register (n=0–1) 1357
Offset: Port0: 0x000A0758, Port1: 0x000A0F58
Table 1820: Low Frequency Configuration Register (n=0–1) 1357
Offset: Port0: 0x000A075C, Port1: 0x000A0F5C
Table 1821: Block Alignment Good Block Weight Register (n=0–1) 1357
Offset: Port0: 0x000A0760, Port1: 0x000A0F60
Table 1822: Block Alignment Aligned EIEOS Weight Register (n=0–1) 1358
Offset: Port0: 0x000A0764, Port1: 0x000A0F64
Table 1823: Block Alignment Bad Block Weight Register (n=0–1) 1358
Offset: Port0: 0x000A0768, Port1: 0x000A0F68
Table 1824: Block Alignment Misaligned EIEOS Weight Register (n=0–1) 1358
Offset: Port0: 0x000A076C, Port1: 0x000A0F6C
Table 1825: Block Alignment Squelch Detect Register (n=0–1) 1358
Offset: Port0: 0x000A0770, Port1: 0x000A0F70
Table 1826: BIST Control Input Register (n=0–1) 1359
Offset: Port0: 0x000A0780, Port1: 0x000A0F80
Table 1827: BIST Configuration Input Register (n=0–1) 1359
Offset: Port0: 0x000A0784, Port1: 0x000A0F84
Table 1828: BIST Data Input Register (n=0–1) 1360
Offset: Port0: 0x000A0788, Port1: 0x000A0F88
Table 1829: BIST LANE and TYPE Register (n=0–1) 1360
Offset: Port0: 0x000A078C, Port1: 0x000A0F8C
Table 1830: BIST Loggin Window Start Cycle Register (n=0–1) 1360
Offset: Port0: 0x000A0790, Port1: 0x000A0F90
Table 1831: BIST Loggin Window Length Register (n=0–1) 1360
Offset: Port0: 0x000A0794, Port1: 0x000A0F94
Table 1832: BIST Result Mask Low Register (n=0–1) 1361
Offset: Port0: 0x000A0798, Port1: 0x000A0F98
Table 1833: BIST Result Mask High Register (n=0–1) 1361
Offset: Port0: 0x000A079C, Port1: 0x000A0F9C
Table 1834: BIST Result Low Register (n=0–1) 1361
Offset: Port0: 0x000A07A0, Port1: 0x000A0FA0
Table 1835: BIST Result High Register (n=0–1) 1361
Offset: Port0: 0x000A07A4, Port1: 0x000A0FA4
Table 1836: BIST Sequencer Configuration Register (n=0–1) 1361
Offset: Port0: 0x000A07A8, Port1: 0x000A0FA8
Table 1837: BIST LFSR Seed Register (n=0–1) 1362
Offset: Port0: 0x000A07AC, Port1: 0x000A0FAC

A.25 XOR Engine Registers 1363

Table 1839: XOR Engine Window Control (XEnWCR)<n> Register (m=0–1, n=0–1) 1366
Offset: Xor Engine0XOR Channel0: 0x00060A40, Xor Engine0XOR Channel1: 0x00060A44, Xor Engine1XOR Channel0: 0x00060B40, Xor Engine1XOR Channel1: 0x00060B44
Table 1840: XOR Engine Base Address (XEBARx)<n> Register (m=0–1, n=0–7) 1367
Offset: Xor Engine0XEBAR0: 0x00060A50, Xor Engine0XEBAR1: 0x00060A54...Xor Engine1XEBAR7: 0x00060B6C
Table 1841: XOR Engine Size Mask (XESMRx)<n> Register (m=0–1, n=0–7) 1367
Offset: Xor Engine0XESMR0: 0x00060A70, Xor Engine0XESMR1: 0x00060A74...Xor Engine1XESMR7: 0x00060B8C

Table 1842: XOR Engine High Address Remap (XEHARRx)<n> Register (m=0–1, n=0–3) 1368
Offset: Xor Engine0XEHARR0: 0x00060A90, Xor Engine0XEHARR1: 0x00060A94, Xor Engine0XEHARR2: 0x00060A98, Xor Engine0XEHARR3: 0x00060A9C, Xor Engine1XEHARR0: 0x00060B90, Xor Engine1XEHARR1: 0x00060B94, Xor Engine1XEHARR2: 0x00060B98, Xor Engine1XEHARR3: 0x00060B9C
Table 1843: XOR Engine Address Override Control (XEAOCR)<n> Register (m=0–1, n=0–1) 1368
Offset: Xor Engine0XEAOCR0: 0x00060AA0, Xor Engine0XEAOCR1: 0x00060AA4, Xor Engine1XEAOCR0: 0x00060BA0, Xor Engine1XEAOCR1: 0x00060BA4
Table 1844: XOR Engine Channel Arbiter (XECHAR) Register (m=0–1) 1370
Offset: Xor Engine0: 0x00060800, Xor Engine1: 0x00060900
Table 1845: XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) 1371
Offset: Xor Engine0XOR Channel0: 0x00060810, Xor Engine0XOR Channel1: 0x00060814, Xor Engine1XOR Channel0: 0x00060910, Xor Engine1XOR Channel1: 0x00060914
Table 1846: XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) 1372
Offset: Xor Engine0XOR Channel0: 0x00060820, Xor Engine0XOR Channel1: 0x00060824, Xor Engine1XOR Channel0: 0x00060920, Xor Engine1XOR Channel1: 0x00060924
Table 1847: XOR Engine Outstanding Reads (XEnORDS)<n> Register (m=0–1, n=0–1) 1373
Offset: Xor Engine0XOR Channel0: 0x00060880, Xor Engine0XOR Channel1: 0x00060884, Xor Engine1XOR Channel0: 0x00060980, Xor Engine1XOR Channel1: 0x00060984
Table 1848: XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1) 1374
Offset: Xor Engine0XOR Channel0: 0x00060A00, Xor Engine0XOR Channel1: 0x00060A04, Xor Engine1XOR Channel0: 0x00060B00, Xor Engine1XOR Channel1: 0x00060B04
Table 1849: XOR Engine Current Descriptor Pointer (XEnCDPR)<n> Register (m=0–1, n=0–1) 1374
Offset: Xor Engine0XOR Channel0: 0x00060A10, Xor Engine0XOR Channel1: 0x00060A14, Xor Engine1XOR Channel0: 0x00060B10, Xor Engine1XOR Channel1: 0x00060B14
Table 1850: XOR Engine Byte Count (XEnBCR)<n> Register (m=0–1, n=0–1) 1374
Offset: Xor Engine0XOR Channel0: 0x00060A20, Xor Engine0XOR Channel1: 0x00060A24, Xor Engine1XOR Channel0: 0x00060B20, Xor Engine1XOR Channel1: 0x00060B24
Table 1851: XOR Engine Destination Pointer (XEnDPR0)<n> Register (m=0–1, n=0–1) 1375
Offset: Xor Engine0XOR Channel0: 0x00060AB0, Xor Engine0XOR Channel1: 0x00060AB4, Xor Engine1XOR Channel0: 0x00060BB0, Xor Engine1XOR Channel1: 0x00060BB4
Table 1852: XOR Engine Block Size (XEnBSR)<n> Register (m=0–1, n=0–1) 1375
Offset: Xor Engine0XOR Channel0: 0x00060AC0, Xor Engine0XOR Channel1: 0x00060AC4, Xor Engine1XOR Channel0: 0x00060BC0, Xor Engine1XOR Channel1: 0x00060BC4
Table 1853: XOR Engine Timer Mode Control (XETMCR) Register (m=0–1) 1375
Offset: Xor Engine0: 0x00060AD0, Xor Engine1: 0x00060BD0
Table 1854: XOR Engine Timer Mode Initial Value (XETMIVR) Register (m=0–1) 1376
Offset: Xor Engine0: 0x00060AD4, Xor Engine1: 0x00060BD4
Table 1855: XOR Engine Timer Mode Current Value (XETMCVR) Register (m=0–1) 1376
Offset: Xor Engine0: 0x00060AD8, Xor Engine1: 0x00060BD8
Table 1856: XOR Engine Initial Value Low (XEIVRL) Register (m=0–1) 1377
Offset: Xor Engine0: 0x00060AE0, Xor Engine1: 0x00060BE0
Table 1857: XOR Engine Initial Value High (XEIVRH) Register (m=0–1) 1377
Offset: Xor Engine0: 0x00060AE4, Xor Engine1: 0x00060BE4
Table 1858: XOR Engine Interrupt Cause (XEICR1) Register (m=0–1) 1377
Offset: Xor Engine0: 0x00060830, Xor Engine1: 0x00060930
Table 1859: XOR Engine Interrupt Mask (XEIMR) Register (m=0–1) 1380
Offset: Xor Engine0: 0x00060840, Xor Engine1: 0x00060940
Table 1860: XOR Engine Error Cause (XEECR) Register (m=0–1) 1380
Offset: Xor Engine0: 0x00060850, Xor Engine1: 0x00060950

Document Classification: Proprietary Information

Table 1861: XOR Engine Error Address (XEEAR) Register (m=0–1) 1381
Offset: Xor Engine0: 0x00060860, Xor Engine1: 0x00060960

A.26 Multi-Purpose Ports (MPP) Registers 1382

Table 1863: MPP_Control_<n*8>_<n*8+7> Register (n=0–6) 1382
Offset: MPP_Num0: 0x00018000, MPP_Num1: 0x00018004, MPP_Num2: 0x00018008, MPP_Num3: 0x0001800C, MPP_Num4: 0x00018010, MPP_Num5: 0x00018014, MPP_Num6: 0x00018018

Table 1864: MPP Control 56_59 Register 1383
Offset: 0x0001801C

A.27 General Management - DFX Server Registers 1384

Table 1866: Core Divider Clock Control 0 Register 1384
Offset: 0x000E4250

Table 1867: Core Divider Clock Control 1 Register 1385
Offset: 0x000E4254

Table 1868: Core Divider Clock Control 2 Register 1385
Offset: 0x000E4258

Table 1869: Core Divider Clock Control 3 Register 1385
Offset: 0x000E425C

Table 1870: CPU PLL Clock Divider Control 0 Register 1386
Offset: 0x000E4260

Table 1871: CPU PLL Clock Divider Control 1 Register 1387
Offset: 0x000E4264

Table 1872: Server eFuse Control Register 1387
Offset: 0x000E4008

Table 1873: Server Reset Control Register 1388
Offset: 0x000E400C

Table 1874: PCIe Skip Initialization Matrix Register 1388
Offset: 0x000E406C

Table 1875: Temperature Sensor 28nm Control LSB Register 1388
Offset: 0x000E4070

Table 1876: Temperature Sensor 28nm Control MSB Register 1389
Offset: 0x000E4074

Table 1877: Temperature Sensor Status Register 1390
Offset: 0x000E4078

Table 1878: AVS Debug Control Register 1390
Offset: 0x000E4124

Table 1879: AVS Enabled Control Register 1390
Offset: 0x000E4130

Table 1880: Device ID Status Register 1390
Offset: 0x000E4240

Table 1881: JTAG Device ID Status Register 1391
Offset: 0x000E4244

Table 1882: DDR-PLL Parameters Register 1391
Offset: 0x000E42F0

Copyright © 2015 Marvell

A.28 Clock Complex Registers 1392
Table 1884: CPU Divider Clock Control Ratio Register 1392
Offset: 0x00018704
Table 1885: TDM DCO Divider Clock Control Register 1392
Offset: 0x00018730

A.29 System Registers 1393
Table 1887: SoC Control 0 Register 1394
Offset: 0x00018200
Table 1888: SoC Control 1 Register 1394
Offset: 0x00018204
Table 1889: SoC Device Multiplex Register 1396
Offset: 0x00018208
Table 1890: Device ID Register 1397
Offset: 0x00018238
Table 1891: Device and Version ID Register 1397
Offset: 0x0001823C
Table 1892: SYSRSTn Length Counter Register 1397
Offset: 0x00018250
Table 1893: CPU Parity Fail Error Register 1398
Offset: 0x00018254
Table 1894: RSTOUTn Mask Register 1399
Offset: 0x00018260
Table 1895: System Soft Reset Register 1400
Offset: 0x00018264
Table 1896: Device Capabilities 0 Register 1402
Offset: 0x00018284
Table 1897: Device Capabilities 1 Register 1402
Offset: 0x00018288
Table 1898: PCI Express Boot Address Register 1403
Offset: 0x000182D4
Table 1899: General Purpose Reserved0 Register 1403
Offset: 0x000182E0
Table 1900: TDM PLL Configuration 0 Register 1403
Offset: 0x00018400
Table 1901: TDM PLL Configuration 1 Register 1404
Offset: 0x00018404
Table 1902: TDM PLL Configuration 2 Register 1405
Offset: 0x00018408
Table 1903: Audio PLL Configuration 0 Register 1405
Offset: 0x00018410
Table 1904: Audio PLL Configuration 1 Register 1407
Offset: 0x00018414
Table 1905: Audio PLL Configuration 2 Register 1407
Offset: 0x00018418
Table 1906: USB Configuration Register 1408
Offset: 0x00018420

Copyright © 2015 Marvell

Table 1907: UTMI PHY<n> Configuration Status Register (n=0–2) 1408
Offset: PHY_NUM0: 0x00018440, PHY_NUM1: 0x00018444, PHY_NUM2: 0x00018448
Table 1908: SDIO3 Configuration Register 1409
Offset: 0x00018454
Table 1909: QSGMII Control 1 Register 1409
Offset: 0x00018494
Table 1910: QSGMII Status Register 1410
Offset: 0x00018498
Table 1911: RTC SoC Interrupt Register 1411
Offset: 0x000184A8
Table 1912: SSCG Frequency Configuration Register 1411
Offset: 0x000184AC
Table 1913: SSCG SSC Mode Register 1411
Offset: 0x000184B0
Table 1914: LDO Control Register 1412
Offset: 0x000184C0
Table 1915: IO Configuration 1 Register 1413
Offset: 0x000184E4
Table 1916: IO Configuration 2 Register 1413
Offset: 0x000184E8
Table 1917: PCIe01 Reference Clock Buffer Control Register 1414
Offset: 0x000184F0
Table 1918: IO Configuration 4 Register 1414
Offset: 0x000184F4
Table 1919: Power Management Clock Gating Control Register 1414
Offset: 0x00018220
Table 1920: BootROM Routine and Error Code Register 1416
Offset: 0x000182D0
Table 1921: Misc Error Interrupt Cause Register 1418
Offset: 0x000182A8
Table 1922: Misc Error Interrupt Mask Register 1419
Offset: 0x000182AC
Table 1923: Sample at Reset (SAR) Register 1420
Offset: 0x00018600
Table 1924: PUP Enable Register 1422
Offset: 0x0001864C
Table 1925: On-Die Pattern Generator (ODPG) Enable Register 1422
Offset: 0x000186D4

Copyright © 2015 Marvell

A ARMADA® 38x Family Register Set

A.1 Registers Overview

This section details the device registers.

A.1.1 Register Field Type Codes

The device registers consist of up to 32-bit fields, where each field is associated with one or more bits. Each of these register fields have a unique programming functionality and their operation is defined by the field's type. The following list describes the function of each type:

Table 91: Standard Register Field Type Codes

Type	Description
LH	Register field with latching high function. If status is high, then the register is set to one and remains set until a read operation is performed through the management interface or a reset occurs.
LL	Register field with latching low function. If status is low, then the register is cleared to zero and remains cleared until a read operation is performed through the management interface or a reset occurs.
Retain	The register value is retained after software reset is executed.
RO	Read Only. Writing to this type of field may cause unpredictable results.
ROC	Read Only Clear. After read, register field is cleared to zero.
RSVD	Reserved for future use. All reserved bits are read as zero unless otherwise noted.
RW	Read and Write.
RW0C	Read-only status, Write-0 to clear status register. Register bits indicate status when read, a set bit indicates a status event may be cleared by writing a 0. Writing a 1 to RW0C bits have no effect.
RW1C	Read-only status, Write-1 to clear status register. Register bits indicate status when read, a set bit indicates a status event may be cleared by writing a 1. Writing a 0 to RW1C bits have no effect.
SC	Self-Clear. Writing a one to this register causes the desired function to be immediately executed, then the register field is cleared to zero when the function is complete.
Special	Used in special cases where the register type functionality does not conform to any of the standard types listed in this table. Refer to the Description Column of the register table for a full definition of the field/register type and functionality.
Update	Value written to the register field does not take effect until a software reset is executed. The value can still be read after it is written.
WO	Write Only. A write to the register field will trigger an internal function and a read will return an undefined value.
X	These bits do not exist.

Copyright © 2015 Marvell

A.2 CPU Subsystem Registers

The following table provides a summarized list of all registers that belong to the CPU Subsystem, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 92: Summary Map Table for the CPU Subsystem Registers

Register Name	Offset	Table and Page
CPU Subsystem Control and Configuration		
SoC Address Map		
IO Window 0 Control Register	0x00020000	Table 93, p. 445
IO Window 0 Base Register	0x00020004	Table 94, p. 445
IO Window 0 Remap Low Register	0x00020008	Table 95, p. 446
IO Window 0 Remap High Register	0x0002000C	Table 96, p. 446
IO Window 1 Control Register	0x00020010	Table 97, p. 446
IO Window 1 Base Register	0x00020014	Table 98, p. 447
IO Window 1 Remap Low Register	0x00020018	Table 99, p. 447
IO Window 1 Remap High Register	0x0002001C	Table 100, p. 447
IO Window 2 Control Register	0x00020020	Table 101, p. 447
IO Window 2 Base Register	0x00020024	Table 102, p. 448
IO Window 2 Remap Low Register	0x00020028	Table 103, p. 448
IO Window 2 Remap High Register	0x0002002C	Table 104, p. 448
IO Window 3 Control Register	0x00020030	Table 105, p. 448
IO Window 3 Base Register	0x00020034	Table 106, p. 449
IO Window 3 Remap Low Register	0x00020038	Table 107, p. 449
IO Window 3 Remap High Register	0x0002003C	Table 108, p. 449
IO Window 4 Control Register	0x00020040	Table 109, p. 449
IO Window 4 Base Register	0x00020044	Table 110, p. 450
IO Window 4 Remap Low Register	0x00020048	Table 111, p. 450
IO Window4 Remap High Register	0x0002004C	Table 112, p. 450
IO Window 5 Control Register	0x00020050	Table 113, p. 450
IO Window 5 Base Register	0x00020054	Table 114, p. 451
IO Window 5 Remap Low Register	0x00020058	Table 115, p. 451
IO Window 5 Remap High Register	0x0002005C	Table 116, p. 451
IO Window 6 Control Register	0x00020060	Table 117, p. 451
IO Window 6 Base Register	0x00020064	Table 118, p. 452
IO Window 6 Remap Low Register	0x00020068	Table 119, p. 452
IO Window 6 Remap High Register	0x0002006C	Table 120, p. 452
IO Window 7 Control Register	0x00020070	Table 121, p. 452
IO Window 7 Base Register	0x00020074	Table 122, p. 453
IO Window 7 Remap Low Register	0x00020078	Table 123, p. 453
IO Window 7 Remap High Register	0x0002007C	Table 124, p. 453
Internal Registers Base Address	0x00020080	Table 125, p. 453

Copyright © 2015 Marvell

Table 92: Summary Map Table for the CPU Subsystem Registers (Continued)

Register Name	Offset	Table and Page
Internal Units Sync Barrier Control Register	0x00020084	Table 126, p. 453
IO Window 8 Control Register	0x00020090	Table 127, p. 454
IO Window 8 Base Register	0x00020094	Table 128, p. 454
IO Window 9 Control Register	0x00020098	Table 129, p. 454
IO Window 9 Base Register	0x0002009C	Table 130, p. 455
IO Window 10 Control Register	0x000200A0	Table 131, p. 455
IO Window 10 Base Register	0x000200A4	Table 132, p. 456
IO Window 11 Control Register	0x000200A8	Table 133, p. 456
IO Window 11 Base Register	0x000200AC	Table 134, p. 456
IO Window 12 Control Register	0x000200B0	Table 135, p. 456
IO Window 12 Base Register	0x000200B4	Table 136, p. 457
IO Window 13 Control Register	0x000200B8	Table 137, p. 457
IO Window 13 Base Register	0x000200BC	Table 138, p. 458
IO Window 14 Control Register	0x000200C0	Table 139, p. 458
IO Window 14 Base Register	0x000200C4	Table 140, p. 458
IO Window 15 Control Register	0x000200C8	Table 141, p. 458
IO Window 15 Base Register	0x000200CC	Table 142, p. 459
IO Window 16 Control Register	0x000200D0	Table 143, p. 459
IO Window 16 Base Register	0x000200D4	Table 144, p. 459
IO Window 17 Control Register	0x000200D8	Table 145, p. 460
IO Window 17 Base Register	0x000200DC	Table 146, p. 460
IO Window 18 Control Register	0x000200E0	Table 147, p. 460
IO Window 18 Base Register	0x000200E4	Table 148, p. 461
IO Window 19 Control Register	0x000200E8	Table 149, p. 461
IO Window 19 Base Register	0x000200EC	Table 150, p. 461
IO Window 13 Remap Low Register	0x000200F0	Table 151, p. 462
IO Window 13 Remap High Register	0x000200F4	Table 152, p. 462
DRAM Window 0 Base Address Register	0x00020180	Table 153, p. 462
DRAM Window 0 Control Register	0x00020184	Table 154, p. 462
DRAM Window 1 Base Address Register	0x00020188	Table 155, p. 463
DRAM Window 1 Control Register	0x0002018C	Table 156, p. 463
DRAM Window 2 Base Address Register	0x00020190	Table 157, p. 464
DRAM Window 2 Control Register	0x00020194	Table 158, p. 464
DRAM Window 3 Base Address Register	0x00020198	Table 159, p. 465
DRAM Window 3 Control Register	0x0002019C	Table 160, p. 465
SoC Window Control Register	0x00020250	Table 161, p. 465
SoC Window Base Register	0x00020254	Table 162, p. 466
Coherency Fabric Control and Status		
IO Error Control Register	0x00020200	Table 163, p. 466
PD Adapter Configuration Register	0x00020220	Table 164, p. 466
CPU Subsystem Error Status Register	0x00020258	Table 165, p. 467

Document Classification: Proprietary Information

Table 92: Summary Map Table for the CPU Subsystem Registers (Continued)

Register Name	Offset	Table and Page
CPU Subsystem Error Mask Register	0x0002025C	Table 166, p. 469
CPU Subsystem Local Cause Register	0x00020260	Table 167, p. 469
Coherency IO Bridge Control and Status		
CIB Control and Configuration Register	0x00020280	Table 168, p. 469
CIB Non L1 Snoopable Window 0 Control Register	0x00020284	Table 169, p. 471
CIB Non L1 Snoopable Window 0 Base Address Register	0x00020288	Table 170, p. 471
CIB Non L1 Snoopable Window 1 Control Register	0x0002028C	Table 171, p. 471
CIB Non L1 Snoopable Window 1 Base Address Register	0x00020290	Table 172, p. 471
CIB Read Buffer Select Register	0x00020294	Table 173, p. 472
CIB Parity Error Status Register	0x00020298	Table 174, p. 472
SoC Global Timers		
SoC Global Timers Control Register	0x00020300	Table 175, p. 472
SoC Global Timers Events Status register Register	0x00020304	Table 176, p. 475
SoC Global Timer 0 Reload Register	0x00020310	Table 177, p. 475
SoC Global Timer 0 Register	0x00020314	Table 178, p. 476
SoC Global Timer 1 Reload Register	0x00020318	Table 179, p. 476
SoC Global Timer 1 Register	0x0002031C	Table 180, p. 476
SoC Global Timer 2 Reload Register	0x00020320	Table 181, p. 476
SoC Global Timer 2 Register	0x00020324	Table 182, p. 477
SoC Global Timer 3 Reload Register	0x00020328	Table 183, p. 477
SoC Global Timer 3 Register	0x0002032C	Table 184, p. 477
Shared Inbound Doorbell		
Shared Inbound Doorbell0 Register	0x00020400	Table 185, p. 477
Shared Inbound Doorbell1 Register	0x00020404	Table 186, p. 477
Shared Inbound Doorbell2 Register	0x00020408	Table 187, p. 478
SoC Attribute Map		
Mbus Units Priority Control Register	0x00020420	Table 188, p. 478
Fabric Units Priority Control Register	0x00020424	Table 189, p. 480
Mbus Units Prefetch Control Register	0x00020428	Table 190, p. 480
MMU Static Windows		
Memory Management Unit (MMU) Static WindowN Register (N=0–7)	Windows0: 0x00020890, Windows1: 0x00020894, Windows2: 0x00020898, Windows3: 0x0002089C, Windows4: 0x000208A0, Windows5: 0x000208A4, Windows6: 0x000208A8, Windows7: 0x000208AC	Table 191, p. 482
Semaphores		
Semaphore0 Register	0x00020500	Table 192, p. 482

Table 92: Summary Map Table for the CPU Subsystem Registers (Continued)

Register Name	Offset	Table and Page
Semaphoren Register (n=1–7)	Semaphore1: 0x00020504, Semaphore2: 0x00020508, Semaphore3: 0x0002050C, Semaphore4: 0x00020510, Semaphore5: 0x00020514, Semaphore6: 0x00020518, Semaphore7: 0x0002051C	Table 193, p. 483
Global Scratch Pad i Register (i=0–7)	pad id0: 0x00020520, pad id1: 0x00020540, pad id2: 0x00020560, pad id3: 0x00020580, pad id4: 0x000205A0, pad id5: 0x000205C0, pad id6: 0x000205E0, pad id7: 0x00020600	Table 194, p. 484
Clock and Reset Control		
Watchdog (WD) RSTOUTn Mask Register	0x00020704	Table 195, p. 484
Watchdog (WD) Interrupt Mask Register	0x00020708	Table 196, p. 484
CPUn Software Reset Control Register (n=0–1)	CPU ID0: 0x00020800, CPU ID1: 0x00020808	Table 197, p. 485
Endpoint Interrupt Controller		
Endpoint Interrupt Cause Vec0 Register	0x00020900	Table 198, p. 485
Endpoint Interrupt Cause Vec1 Register	0x00020904	Table 199, p. 485
Endpoint Interrupt Cause Vec2 Register	0x00020908	Table 200, p. 486
Endpoint Interrupt Cause Vec3 Register	0x0002090C	Table 201, p. 486
Endpoint Select Cause Register	0x00020928	Table 202, p. 486
Endpoint Fabric Local Interrupt Mask Register	0x00020940	Table 203, p. 487
Endpoint SoC Main Interrupt Error 0 Mask Register	0x00020944	Table 204, p. 487
Endpoint SoC Main Interrupt Error 1 Mask Register	0x00020948	Table 205, p. 487
Main Interrupt Controller		
Software Triggered Interrupt Register	0x00020A04	Table 206, p. 488
SoC Main Interrupt Error 0 Cause Register	0x00020A20	Table 207, p. 489
SoC Main Interrupt Error 1 Cause Register	0x00020A24	Table 208, p. 489
Endpoint ISE Register	0x00020A30	Table 209, p. 489
Endpoint ICE Register	0x00020A34	Table 210, p. 489
Interrupt Source i Control Register (i=0–28)	Interrupt ID0: 0x00020B00, Interrupt ID1: 0x00020B04...Interrupt ID28: 0x00020B70	Table 211, p. 490
Endpoint Source i Control Register (i=29–115)	Interrupt ID29: 0x00020B74, Interrupt ID30: 0x00020B78...Interrupt ID115: 0x00020CCC	Table 212, p. 491
CoreSight Control		
CoreSight Components Base Address Register	0x00020D00	Table 213, p. 492
CoreSight Status Register	0x00020D0C	Table 214, p. 492
CA9 MP Core SoC Control		

Document Classification: Proprietary Information

Table 92: Summary Map Table for the CPU Subsystem Registers (Continued)

Register Name	Offset	Table and Page
CA9 MP Core Reset Control Register	0x00020D84	Table 215, p. 492
CPU Local		
CPU Configuration Register (N=0–1)	CPU ID0: 0x00021800, CPU ID1: 0x00021900	Table 216, p. 493
CPU Control and Status Register (N=0–1)	CPU ID0: 0x00021808, CPU ID1: 0x00021908	Table 217, p. 494
CPU IO Sync Barrier Control Register (N=0–1)	CPU ID0: 0x00021810, CPU ID1: 0x00021910	Table 218, p. 495
SoC Private Timers Control Register (N=0–1)	CPU ID0: 0x00021840, CPU ID1: 0x00021940	Table 219, p. 495
SoC Private Timer 0 Reload Register (N=0–1)	CPU ID0: 0x00021850, CPU ID1: 0x00021950	Table 220, p. 496
SoC Private Timer 0 Register (N=0–1)	CPU ID0: 0x00021854, CPU ID1: 0x00021954	Table 221, p. 497
SoC Private Timer 1 Reload Register (N=0–1)	CPU ID0: 0x00021858, CPU ID1: 0x00021958	Table 222, p. 497
SoC Private Timer 1 Register (N=0–1)	CPU ID0: 0x0002185C, CPU ID1: 0x0002195C	Table 223, p. 497
SoC Private Timers Events Status Register (N=0–1)	CPU ID0: 0x00021868, CPU ID1: 0x00021968	Table 224, p. 497
Outbound Doorbell Register (N=0–1)	CPU ID0: 0x00021870, CPU ID1: 0x00021970	Table 225, p. 498
Outbound Doorbell Mask Register (N=0–1)	CPU ID0: 0x00021874, CPU ID1: 0x00021974	Table 226, p. 498
Inbound Doorbell Register (N=0–1)	CPU ID0: 0x00021878, CPU ID1: 0x00021978	Table 227, p. 499
Inbound Doorbell Mask Register (N=0–1)	CPU ID0: 0x0002187C, CPU ID1: 0x0002197C	Table 228, p. 499
SoC PPI Cause Reg Register (N=0–1)	CPU ID0: 0x00021880, CPU ID1: 0x00021980	Table 229, p. 499
SoC Private Peripheral Interrupt Set Mask (PPI ISM) Register (N=0–1)	CPU ID0: 0x000218B8, CPU ID1: 0x000219B8	Table 230, p. 500
SoC Private Peripheral Interrupt Clear Mask (PPI ICM) Register (N=0–1)	CPU ID0: 0x000218BC, CPU ID1: 0x000219BC	Table 231, p. 500
SoC Main Interrupt Error 0 Mask Register (N=0–1)	CPU ID0: 0x000218C0, CPU ID1: 0x000219C0	Table 232, p. 500
SoC Main Interrupt Error 1 Mask Register (N=0–1)	CPU ID0: 0x000218C8, CPU ID1: 0x000219C8	Table 233, p. 501
Local Scratch Pad 0 Register (N=0–1)	CPU ID0: 0x000218D0, CPU ID1: 0x000219D0	Table 234, p. 501
Local Scratch Pad 1 Register (N=0–1)	CPU ID0: 0x000218D4, CPU ID1: 0x000219D4	Table 235, p. 501
Local Scratch Pad 2 Register (N=0–1)	CPU ID0: 0x000218D8, CPU ID1: 0x000219D8	Table 236, p. 501
Local Scratch Pad 3 Register (N=0–1)	CPU ID0: 0x000218DC, CPU ID1: 0x000219DC	Table 237, p. 501
Local Scratch Pad 4 Register (N=0–1)	CPU ID0: 0x000218E0, CPU ID1: 0x000219E0	Table 238, p. 502

Copyright © 2015 Marvell

Table 92: Summary Map Table for the CPU Subsystem Registers (Continued)

Register Name	Offset	Table and Page
Local Scratch Pad 5 Register (N=0–1)	CPU ID0: 0x000218E4, CPU ID1: 0x000219E4	Table 239, p. 502
Local Scratch Pad 6 Register (N=0–1)	CPU ID0: 0x000218E8, CPU ID1: 0x000219E8	Table 240, p. 502
Local Scratch Pad 7 Register (N=0–1)	CPU ID0: 0x000218EC, CPU ID1: 0x000219EC	Table 241, p. 502
CPU Status Register (N=0–1)	CPU ID0: 0x000218F0, CPU ID1: 0x000219F0	Table 242, p. 502
CPU External Debug Register (N=0–1)	CPU ID0: 0x000218F4, CPU ID1: 0x000219F4	Table 243, p. 503
CPU Data Engine Flags Register (N=0–1)	CPU ID0: 0x000218F8, CPU ID1: 0x000219F8	Table 244, p. 503
CPU Event Control Register (N=0–1)	CPU ID0: 0x000218FC, CPU ID1: 0x000219FC	Table 245, p. 503
Timestamp		
Timestamp Counter Enable Register	0x00023000	Table 246, p. 504
Timestamp Counter Preload 1 Register	0x00023004	Table 247, p. 504
Timestamp Counter Preload 2 Register	0x00023008	Table 248, p. 504
Timestamp Claim Tag Set Register	0x00023FA0	Table 249, p. 504
Timestamp Claim Tag Clear Register	0x00023FA4	Table 250, p. 504
Timestamp Lock Access Register	0x00023FB0	Table 251, p. 505
Timestamp Lock Status Register	0x00023FB4	Table 252, p. 505
Timestamp Authentication Status Register	0x00023FB8	Table 253, p. 505
Timestamp Device ID Register	0x00023FC8	Table 254, p. 505
Timestamp Device Type Identifier Register	0x00023FCC	Table 255, p. 505
Timestamp Peripheral ID4 Register	0x00023FD0	Table 256, p. 505
Timestamp Peripheral ID5 Register	0x00023FD4	Table 257, p. 506
Timestamp Peripheral ID6 Register	0x00023FD8	Table 258, p. 506
Timestamp Peripheral ID7 Register	0x00023FDC	Table 259, p. 506
Timestamp Peripheral ID0 Register	0x00023FE0	Table 260, p. 506
Timestamp Peripheral ID1 Register	0x00023FE4	Table 261, p. 506
Timestamp Peripheral ID2 Register	0x00023FE8	Table 262, p. 506
Timestamp Peripheral ID3 Register	0x00023FEC	Table 263, p. 507
Timestamp Component ID0 Register	0x00023FF0	Table 264, p. 507
Timestamp Component ID1 Register	0x00023FF4	Table 265, p. 507
Timestamp Component ID2 Register	0x00023FF8	Table 266, p. 507
Timestamp Component ID3 Register	0x00023FFC	Table 267, p. 507

Document Classification: Proprietary Information

A.2.1 CPU Subsystem Control and Configuration

A.2.1.1 SoC Address Map

Table 93: IO Window 0 Control Register
Offset: 0x00020000

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 64 KB granularity (for example, a value of 0x00FF specifies 256*64KB = 16 MB). NOTES: A value of 0x0 specifies 64-KB size. Changing the value of this field must be done when <win_en>=0.
15:8	Attr	RW 0xE8	Specifies the target interface attributes associated with this window. See the "Target Interface Configurations" section. Changing the value of this field must be done when <win_en>=0.
7:4	Target	RW 0x4	Specifies the target interface associated with this window. See the "Target Interface Configurations" section. Changing the value of this field must be done when <win_en>=0.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	Window Sync Barrier Enable When enabled, data associated with read requests that hit the window will be returned to the requesting CPU after all writes issued by the target unit reach the memory. 0 = False 1 = True
0	win_en	RW 0x1	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Table 94: IO Window 0 Base Register
Offset: 0x00020004

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register!			
31:16	Base	RW 0x8000	Base Address Used with the <Size> field in the Window0 Control Register to set the address window size and location. Corresponds to transaction address[31:16]. Changing the value of this field must be done when <win0_en>=0.

Table 94: IO Window 0 Base Register (Continued)
Offset: 0x00020004

Bit	Field	Type/InitVal	Description
15:4	Reserved	RO 0x0	Reserved
3:0	Reserved	RSVD 0x0	Reserved

Table 95: IO Window 0 Remap Low Register
Offset: 0x00020008

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x8000	Remap Address Used with the <Size> field in the Window0 Control Register to specifies address bits[31:0] to be driven to the target interface. Changing the value of this field must be done when <win0_en>=0.
15:0	Reserved	RO 0x0	Reserved

Table 96: IO Window 0 Remap High Register
Offset: 0x0002000C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Specifies address bits[63:32] to be driven to the target interface.

Table 97: IO Window 1 Control Register
Offset: 0x00020010

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register
15:8	Attr	RW 0xE8	See the Window0 Control Register.
7:4	Target	RW 0x4	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 98: IO Window 1 Base Register
Offset: 0x00020014

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register.			
31:16	Base	RW 0x8800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 99: IO Window 1 Remap Low Register
Offset: 0x00020018

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x8800	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 100: IO Window 1 Remap High Register
Offset: 0x0002001C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 101: IO Window 2 Control Register
Offset: 0x00020020

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register.
15:8	Attr	RW 0xE8	See the Window0 Control Register.
7:4	Target	RW 0x4	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 102: IO Window 2 Base Register
Offset: 0x00020024

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window2 Remap Low Register must be set to same value as the <Base> field in this register.			
31:16	Base	RW 0x9000	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 103: IO Window 2 Remap Low Register
Offset: 0x00020028

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x9000	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 104: IO Window 2 Remap High Register
Offset: 0x0002002C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 105: IO Window 3 Control Register
Offset: 0x00020030

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register.
15:8	Attr	RW 0xE8	See the Window0 Control Register.
7:4	Target	RW 0x4	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 106: IO Window 3 Base Register
Offset: 0x00020034

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window3 Remap Low Register must be set to same value as the <Base> field in this register.			
31:16	Base	RW 0x9800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 107: IO Window 3 Remap Low Register
Offset: 0x00020038

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x9800	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 108: IO Window 3 Remap High Register
Offset: 0x0002003C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 109: IO Window 4 Control Register
Offset: 0x00020040

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register.
15:8	Attr	RW 0xE8	See the Window0 Control Register.
7:4	Target	RW 0x8	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 110: IO Window 4 Base Register
Offset: 0x00020044

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xA000	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 111: IO Window 4 Remap Low Register
Offset: 0x00020048

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0xA000	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 112: IO Window4 Remap High Register
Offset: 0x0002004C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 113: IO Window 5 Control Register
Offset: 0x00020050

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register.
15:8	Attr	RW 0xE8	See the Window0 Control Register.
7:4	Target	RW 0x8	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x0	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 114: IO Window 5 Base Register
Offset: 0x00020054

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xA800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 115: IO Window 5 Remap Low Register
Offset: 0x00020058

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0xA800	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 116: IO Window 5 Remap High Register
Offset: 0x0002005C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 117: IO Window 6 Control Register
Offset: 0x00020060

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register
15:8	Attr	RW 0xE8	See the Window0 Control Register.
7:4	Target	RW 0x8	See the Window0 Control Register.
3:2	Reserved	RO 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x0	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 118: IO Window 6 Base Register
Offset: 0x00020064

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xB000	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 119: IO Window 6 Remap Low Register
Offset: 0x00020068

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0xB000	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 120: IO Window 6 Remap High Register
Offset: 0x0002006C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 121: IO Window 7 Control Register
Offset: 0x00020070

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register.
15:8	Attr	RW 0xE8	Target specific attributes depending on the target interface. See the Window0 Control Register.
7:4	Target	RW 0x8	See the Window0 Control Register.
3:2	Reserved	RO 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x0	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 122: IO Window 7 Base Register
Offset: 0x00020074

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xB800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 123: IO Window 7 Remap Low Register
Offset: 0x00020078

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0xB800	Remap Address See the Window0 Remap Low Register.
15:0	Reserved	RO 0x0	Reserved

Table 124: IO Window 7 Remap High Register
Offset: 0x0002007C

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap Address See the Window0 Remap High Register.

Table 125: Internal Registers Base Address
Offset: 0x00020080

Bit	Field	Type/InitVal	Description
31:20	Base	RW 0xD00	Internal Registers Base Address
19:0	Reserved	RSVD 0x0	Reserved

Table 126: Internal Units Sync Barrier Control Register
Offset: 0x00020084

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 126: Internal Units Sync Barrier Control Register (Continued)
Offset: 0x00020084

Bit	Field	Type/InitVal	Description
15:0	InternalUnitSyncBarrierEnable	RW 0x0	Internal Unit Sync Barrier Enable This field contains a bit per each Mbus unit. When bit#i in this field is set, the data of a read request from an internal address with (addr[19:16] == i) will be returned to the CPU after all previous write requests from the unit with ID=i have been written to the memory.

Table 127: IO Window 8 Control Register
Offset: 0x00020090

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	See the Window0 Control Register.
15:8	Attr	RW 0x9	See the Window0 Control Register.
7:4	Target	RW 0x9	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 128: IO Window 8 Base Register
Offset: 0x00020094

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xC801	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 129: IO Window 9 Control Register
Offset: 0x00020098

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7FF	See the Window0 Control Register.

Document Classification: Proprietary Information

Table 129: IO Window 9 Control Register (Continued)
Offset: 0x00020098

Bit	Field	Type/InitVal	Description
15:8	Attr	RW 0x2F	See the Window0 Control Register.
7:4	Target	RW 0x1	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 130: IO Window 9 Base Register
Offset: 0x0002009C

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xD800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 131: IO Window 10 Control Register
Offset: 0x000200A0

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	See the Window0 Control Register.
15:8	Attr	RW 0x3E	See the Window0 Control Register.
7:4	Target	RW 0x1	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 132: IO Window 10 Base Register
Offset: 0x000200A4

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xE000	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 133: IO Window 11 Control Register
Offset: 0x000200A8

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	See the Window0 Control Register.
15:8	Attr	RW 0x3D	See the Window0 Control Register.
7:4	Target	RW 0x1	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 134: IO Window 11 Base Register
Offset: 0x000200AC

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xE800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 135: IO Window 12 Control Register
Offset: 0x000200B0

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	See the Window0 Control Register.

Document Classification: Proprietary Information

Table 135: IO Window 12 Control Register (Continued)
Offset: 0x000200B0

Bit	Field	Type/InitVal	Description
15:8	Attr	RW 0x3B	See the Window0 Control Register.
7:4	Target	RW 0x1	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 136: IO Window 12 Base Register
Offset: 0x000200B4

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xF000	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 137: IO Window 13 Control Register
Offset: 0x000200B8

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	See the Window0 Control Register.
15:8	Attr	RW SAR	See the Window0 Control Register.
7:4	Target	RW 0x1	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 138: IO Window 13 Base Register
Offset: 0x000200BC

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xF800	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 139: IO Window 14 Control Register
Offset: 0x000200C0

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x03FF	See the Window0 Control Register.
15:8	Attr	RW 0x1E	See the Window0 Control Register.
7:4	Target	RW 0x1	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 140: IO Window 14 Base Register
Offset: 0x000200C4

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xD400	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 141: IO Window 15 Control Register
Offset: 0x000200C8

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	See the Window0 Control Register.
15:8	Attr	RW 0xe0	See the Window0 Control Register.

Copyright © 2015 Marvell

Table 141: IO Window 15 Control Register (Continued)
Offset: 0x000200C8

Bit	Field	Type/InitVal	Description
7:4	Target	RW 0x4	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x0	See the Window0 Control Register.

Table 142: IO Window 15 Base Register
Offset: 0x000200CC

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xD030	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 143: IO Window 16 Control Register
Offset: 0x000200D0

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	See the Window0 Control Register.
15:8	Attr	RW 0xe0	See the Window0 Control Register.
7:4	Target	RW 0x4	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x0	See the Window0 Control Register.

Table 144: IO Window 16 Base Register
Offset: 0x000200D4

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xD031	Base Address See the Window0 Base Register.

Copyright © 2015 Marvell

Table 144: IO Window 16 Base Register (Continued)
Offset: 0x000200D4

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 145: IO Window 17 Control Register
Offset: 0x000200D8

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	See the Window0 Control Register.
15:8	Attr	RW 0xe0	See the Window0 Control Register.
7:4	Target	RW 0x4	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x0	See the Window0 Control Register.

Table 146: IO Window 17 Base Register
Offset: 0x000200DC

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xD032	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 147: IO Window 18 Control Register
Offset: 0x000200E0

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x7f	See the Window0 Control Register.
15:8	Attr	RW 0x0	See the Window0 Control Register.
7:4	Target	RW 0xc	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 147: IO Window 18 Control Register (Continued)
Offset: 0x000200E0

Bit	Field	Type/InitVal	Description
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 148: IO Window 18 Base Register
Offset: 0x000200E4

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xD380	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 149: IO Window 19 Control Register
Offset: 0x000200E8

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	See the Window0 Control Register.
15:8	Attr	RW 0x0e	See the Window0 Control Register.
7:4	Target	RW 0x0	See the Window0 Control Register.
3:2	Reserved	RSVD 0x0	Reserved
1	SyncBarrierEnable	RW 0x0	See the Window0 Control Register.
0	win_en	RW 0x1	See the Window0 Control Register.

Table 150: IO Window 19 Base Register
Offset: 0x000200EC

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0000	Base Address See the Window0 Base Register.
15:0	Reserved	RSVD 0x0	Reserved

Table 151: IO Window 13 Remap Low Register
Offset: 0x000200F0

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0xF800	Remap Address Used with the <Size> field in the Window13 Control Register to specify address bits[31:0] to be driven to the target interface. Changing the value of this field must be done when <win13_en>=0.
15:0	Reserved	RO 0x0	Reserved

Table 152: IO Window 13 Remap High Register
Offset: 0x000200F4

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Specifies address bits[63:32] to be driven to the target interface.

Table 153: DRAM Window 0 Base Address Register
Offset: 0x00020180

Bit	Field	Type/InitVal	Description
31:24	Base	RW 0x0	DRAM Window0 base address. Corresponds to address bits [31:24].
23:0	Reserved	RSVD 0x0	Reserved

Table 154: DRAM Window 0 Control Register
Offset: 0x00020184

Bit	Field	Type/InitVal	Description
31:24	Size	RW 0x0f	DRAM Window0 Size Corresponds to Base Address bits [31:24]. Must be programmed from LSb to MSb as a sequence of 1s followed by a sequence of 0s. Changing the value of this field must be done when the window is disabled (<En>=0).
23:5	Reserved	RO 0x7ffff	Reserved

Copyright © 2015 Marvell

Table 154: DRAM Window 0 Control Register (Continued)
Offset: 0x00020184

Bit	Field	Type/InitVal	Description
4:2	Win_CS	RW 0x0	DRAM Window0 CS Defines the DRAM CS that will be accessed upon a hit in DRAM Window 0. Changing the value of this field must be done when the window is disabled (<En>=0) 0 = CS[0]: For DRAM Controller 1 = CS[1]: For DRAM Controller 2 = CS[2]: For DRAM Controller 3 = CS[3]: For DRAM Controller
1	Reserved	RSVD 0x0	Reserved
0	En	RW 0x0	DRAM Window0 Enable 0 = False 1 = True

Table 155: DRAM Window 1 Base Address Register
Offset: 0x00020188

Bit	Field	Type/InitVal	Description
31:24	Base	RW 0x10	DRAM Window1 base address. Corresponds to address bits [31:24].
23:0	Reserved	RSVD 0x0	Reserved

Table 156: DRAM Window 1 Control Register
Offset: 0x0002018C

Bit	Field	Type/InitVal	Description
31:24	Size	RW 0x0f	DRAM Window1 Bank Size Corresponds to Base Address bits [31:24]. Must be programmed from LSb to MSb as a sequence of 1s followed by a sequence of 0s.
23:5	Reserved	RO 0x7ffff	Reserved
4:2	Win_CS	RW 0x1	DRAM Window1 CS Defines the DRAM CS which will be accessed upon a hit in DRAM Window 1. Changing the value of this field must be done when the window is disabled (<En>=0). 0 = CS[0]: For DRAM Controller 1 = CS[1]: For DRAM Controller 2 = CS[2]: For DRAM Controller 3 = CS[3]: For DRAM Controller

Copyright © 2015 Marvell

Table 156: DRAM Window 1 Control Register (Continued)
Offset: 0x0002018C

Bit	Field	Type/InitVal	Description
1	Reserved	RSVD 0x0	Reserved
0	En	RW 0x0	DRAM Window1 Enable 0 = False 1 = True

Table 157: DRAM Window 2 Base Address Register
Offset: 0x00020190

Bit	Field	Type/InitVal	Description
31:24	Base	RW 0x20	DRAM Window2 base address. Corresponds to address bits [31:24].
23:0	Reserved	RSVD 0x0	Reserved

Table 158: DRAM Window 2 Control Register
Offset: 0x00020194

Bit	Field	Type/InitVal	Description
31:24	Size	RW 0x0f	DRAM Window2 Bank Size Corresponds to Base Address bits [31:24]. Must be programmed from LSb to MSb as a sequence of 1s followed by a sequence of 0s.
23:5	Reserved	RO 0x7ffff	Reserved
4:2	Win_CS	RW 0x2	DRAM Window2 CS Defines the DRAM CS which will be accessed upon a hit in DRAM Window 2. Changing the value of this field must be done when the window is disabled (<En>=0). 0 = CS[0]: For DRAM Controller 1 = CS[1]: For DRAM Controller 2 = CS[2]: For DRAM Controller 3 = CS[3]: For DRAM Controller
1	Reserved	RSVD 0x0	Reserved
0	En	RW 0x0	DRAM Window2 Enable 0 = False 1 = True

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 159: DRAM Window 3 Base Address Register
Offset: 0x00020198

Bit	Field	Type/InitVal	Description
31:24	Base	RW 0x30	DRAM Window3 base address. Corresponds to address bits [31:24].
23:0	Reserved	RSVD 0x0	Reserved

Table 160: DRAM Window 3 Control Register
Offset: 0x0002019C

Bit	Field	Type/InitVal	Description
31:24	Size	RW 0x0f	DRAM Window3 Size Corresponds to Base Address bits [31:24]. Must be programmed from LSb to MSb as a sequence of 1s followed by a sequence of 0s.
23:5	Reserved	RO 0x7ffff	Reserved
4:2	Win_CS	RW 0x3	DRAM Window3 CS Defines the DRAM CS that will be accessed upon a hit in DRAM Window 3. Changing the value of this field must be done when the window is disabled (<En>=0). 0 = CS[0]: For DRAM Controller 1 = CS[1]: For DRAM Controller 2 = CS[2]: For DRAM Controller 3 = CS[3]: For DRAM Controller
1	Reserved	RSVD 0x0	Reserved
0	En	RW 0x0	DRAM Window 3 Enable 0 = False 1 = True

Table 161: SoC Window Control Register
Offset: 0x00020250

Bit	Field	Type/InitVal	Description
31:20	Size	RW 0x3FF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 1 MB granularity (for example, a value of 0x00F specifies 16*1MB = 16 MB). **NOTE:** A value of 0x0 specifies 1-MB size.
19:4	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 161: SoC Window Control Register (Continued)
Offset: 0x00020250

Bit	Field	Type/InitVal	Description
3:1	Reserved	RO 0x0	Reserved
0	WinEn	RW 0x1	Mbus Bridge Window Enable 0 = False: Window is disabled. 1 = True: Window is enabled.

Table 162: SoC Window Base Register
Offset: 0x00020254

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register!			
31:20	Base	RW 0x0	Base Address Used with the <Size> field in the Mbus Bridge Window Control register to set the address window size and location. Corresponds to transaction address[31:20].
19:0	Reserved	RSVD 0x0	Reserved

A.2.1.2 Coherency Fabric Control and Status

Table 163: IO Error Control Register
Offset: 0x00020200

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8	IO Err Prop Enable	RW 0x1	Enable MBUS Error Propagation 0 = False: MBUS Error is not propagated to the CPU on read 1 = True: MBUS Error is propagated to the CPU on read
7:0	Reserved	RSVD 0x0	Reserved

Table 164: PD Adapter Configuration Register
Offset: 0x00020220

Bit	Field	Type/InitVal	Description
Punit and Dunit attached adapter control lines.			
31:13	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 164: PD Adapter Configuration Register (Continued)
Offset: 0x00020220

Bit	Field	Type/InitVal	Description
12	DRAMAdapterAsyncMode	RW 0x0	Adapter clock domains are asynchronous. 0 = Synchronous Mode: DRAM Adapter CP and DP clock domains are synchronous. 1 = Asynchronous Mode: DRAM Adapter CP and DP clock domains are asynchronous.
11:0	Reserved	RSVD 0xE0E	Reserved

Table 165: CPU Subsystem Error Status Register
Offset: 0x00020258

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29	CPU1 LSL1 STATE Parity Error	RW0C 0x0	CPU LSL1 State Parity Error Set when a parity error is detected on the LSL1 State memories of CPU1.
28	CPU0 LSL1 STATE Parity Error	RW0C 0x0	CPU LSL1 State Parity Error Set when a parity error is detected on the LSL1 State memories of CPU0.
27:26	Reserved	RSVD 0x0	Reserved
25	CPU1 LSL1 ATTR Parity Error	RW0C 0x0	CPU LSL1 Attr Parity Error Set when a parity error is detected on the LSL1 Attribute memories of CPU1.
24	CPU0 LSL1 ATTR Parity Error	RW0C 0x0	CPU LSL1 Attr Parity Error Set when a parity error is detected on the LSL1 Attribute memories of CPU0.
23:22	Reserved	RSVD 0x0	Reserved
21	CPU1 MMU Parity Error	RW0C 0x0	CPU MMU Parity Error Set when a parity error is detected on the TLB memories of CPU1.
20	CPU0 MMU Parity Error	RW0C 0x0	CPU MMU Parity Error Set when a parity error is detected on the TLB memories of CPU0.
19:18	Reserved	RSVD 0x0	Reserved
17	CPU1 IC Parity Error	RW0C 0x0	CPU's Instruction Cache Parity Error Set when a parity error is detected on the Instruction Cache memories of CPU1.
16	CPU0 IC Parity Error	RW0C 0x0	CPU's Instruction Cache Parity Error Set when a parity error is detected on the Instruction Cache memories of CPU0.
15:14	Reserved	RSVD 0x0	Reserved
13	CPU1 DC Parity Error	RW0C 0x0	CPU's Data Cache Parity Error Set when a parity error is detected on the Data Cache memories of CPU1.

Copyright © 2015 Marvell

Table 165: CPU Subsystem Error Status Register (Continued)
Offset: 0x00020258

Bit	Field	Type/InitVal	Description
12	CPU0 DC Parity Error	RW0C 0x0	CPU's Data Cache Parity Error Set when a parity error is detected on the Data Cache memories of CPU0.
11:10	Reserved	RSVD 0x0	Reserved
9	ETB1 Parity Error	RW0C 0x0	ETB Array Parity Error Set when a parity error has occurred on Embedded Trace Buffer (ETB) memory of CPU1.
8	ETB0 Parity Error	RW0C 0x0	ETB Array Parity Error Set when a parity error has occurred on Embedded Trace Buffer (ETB) memory of CPU0.
7:6	Reserved	RSVD 0x0	Reserved
5	CPU1 Snoop Parity Error	RW0C 0x0	CPU Snoop Parity Error Interrupt Set when a parity error has occurred on snoop to an erroneous Data Cache line in CPU1.
4	CPU0 Snoop Parity Error	RW0C 0x0	CPU Snoop Parity Error Interrupt Set when a parity error has occurred on snoop to an erroneous Data Cache line in CPU0.
3	CIB AccessErr	RW0C 0x0	CIB Access Error Set in the case of a coherent request that is issued by an I/O engine and does not hit an SRAM or a DRAM window.
2	CIB Parity Error	RW0C 0x0	Coherent IO Bridge Parity Error Set when Coherent IO Bridge memories detected a soft error. Details of the parity error are captured at <CIB Parity Error Status>.
1	AccessErr	RW0C 0x0	Access Error. Set in the following error events: * CPU burst access to internal register * CPU access to an unmapped address (no match in any of the address decoding windows)
0	Bit64Err	RW0C 0x0	Mbus Bridge Read Error Set when an erroneous read data indication from the Mbus unit is recognized.

Copyright © 2015 Marvell

Table 166: CPU Subsystem Error Mask Register
Offset: 0x0002025C

Bit	Field	Type/InitVal	Description
31:0	CPUSubsystemErrorMask	RW 0x0	CPU Subsystem Error Mask A mask bit per error event in the CPU Subsystem Error Status Register. When an error event occurs in the CPU Subsystem, the <CPU Subsystem ErrSum> status bit in the CPU Subsystem Local Cause Register is set, if the associated error mask bit is set.

Table 167: CPU Subsystem Local Cause Register
Offset: 0x00020260

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27	CPUSubsystemErrorSum	RO 0x0	CPU Subsystem Error Sum Interrupt
26	Global WD Timer	RO 0x0	Global Watchdog (WD) Timer Interrupt
25:17	Reserved	RSVD 0x0	Reserved
16	L2 Cache Interrupt	RO 0x0	L2 Cache Events Summary Interrupt
15:10	Reserved	RSVD 0x0	Reserved
9	CPU1 nctiirq	RO 0x0	CPU1 Debug Cross Trigger Interupt
8	CPU0 nctiirq	RO 0x0	CPU0 Debug Cross Trigger Interupt
7:2	Reserved	RSVD 0x0	Reserved
1	CPU1 Perf Counter Overflow	RO 0x0	CPU1 Performance Counter Overflow Interrupt
0	CPU0 Perf Counter Overflow	RO 0x0	CPU0 Performance Counter Overflow Interrupt

A.2.1.3 Coherency IO Bridge Control and Status

Table 168: CIB Control and Configuration Register
Offset: 0x00020280

Bit	Field	Type/InitVal	Description
31:21	Reserved	RSVD 0x5FB	Reserved

Table 168: CIB Control and Configuration Register (Continued)
Offset: 0x00020280

Bit	Field	Type/InitVal	Description
20	NonSecureIOTransfer	RW 0x0	Secured/Non secured I/O transfers towards Fabric. 0 = Secured Transfer 1 = Non Secured Transfer
19:14	Reserved	RSVD 0x1C	Reserved
13	CIB Empty Status	RO 0x0	Coherent I/O Bridge Empty Specifies that CIB internal buffers are empty, that is: - All shared commands were sent to Fabric. - All shared writes were completed by Fabric domain. - All shared reads were sent back to the external hosts. 0 = Not Empty 1 = Empty
12	Reserved	RSVD 0x1	Reserved
11:10	IgnoreSharingEnable	RW 0x0	Coherent IO Bridge Ignore Sharing Enable When this bit is set, coherent I/O requests are treated as non-shared, and therefore, ignored by the Coherent I/O Bridge and routed directly to the DRAM controller. This bit is used when the entire CPU Subsystem's caches are flushed to memory (for example, in Power Down state). 0 = Disable: Mbus external shared requests are routed to Coherent I/O Bridge and external non-shared requests are routed to DDR. 2 = Enable: Mbus external shared and external non-shared requests are routed to DDR.
9	StopCibAckEnable	RW 0x0	Stop Coherent I/O Bridge Acknowledge Enable This bit controls the prevention of CIB acknowledgment to pending Mbus shared requests. 0 = False: CIB acknowledges shared read/write pending requests. 1 = True: CIB does not acknowledge shared read/write pending requests, only used as part of power down operation.
8:5	Reserved	RSVD 0x8	Reserved
4:3	Reserved	RW 0x2	Reserved Must be 0x2.
2:1	Reserved	RW 0x3	Reserved Must be 0x3.
0	CIB Write Back Enable	RW 0x1	Coherent I/O Bridge Write Back Mode Enable When this field is set, I/O write requests that are written to the L2 (due to a hit or due to a CL allocation) will not be written to DDR. When this field is cleared, I/O write requests are always written to the DDR. 0 = Disable: CIB in Write Through mode 1 = Enable: CIB in Write Back mode

Copyright © 2015 Marvell

Table 169: CIB Non L1 Snoopable Window 0 Control Register
Offset: 0x00020284

Bit	Field	Type/InitVal	Description
31:16	Window Size	RW 0x0	Non L1 Snoopable Window Size Each bit represents 64 KB size granularity.
15:1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x0	Non L1 Snoopable Window Enable Enable checking I/O requests, if residing within the non L1 snoopable window0 region. If it matches the window address region, no L1 snoop will be applied. 0 = False: CIB non L1 Snoop Window disabled. 1 = True: CIB non L1 Snoop Window enabled.

Table 170: CIB Non L1 Snoopable Window 0 Base Address Register
Offset: 0x00020288

Bit	Field	Type/InitVal	Description
31:16	Window Base	RW 0x0	Non L1 Snoopable Window Base Address Corresponds to transaction address bits [31:16].
15:0	Reserved	RSVD 0x0	Reserved

Table 171: CIB Non L1 Snoopable Window 1 Control Register
Offset: 0x0002028C

Bit	Field	Type/InitVal	Description
31:16	Window Size	RW 0x0	Non L1 Snoopable Window Size Each bit represents 64 KB size granularity.
15:1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x0	Non L1 Snoopable Window Enable Enables checking I/O requests, if residing within non the L1 snoopable window0 region. If it matches the window address region, no L1 snoop will be applied. 0 = False: CIB non L1 Snoop Window disabled. 1 = True: CIB non L1 Snoop Window enabled.

Table 172: CIB Non L1 Snoopable Window 1 Base Address Register
Offset: 0x00020290

Bit	Field	Type/InitVal	Description
31:16	Window Base	RW 0x0	Non L1 Snoopable Window Base Address Corresponds to transaction address bits [31:16].
15:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 173: CIB Read Buffer Select Register
Offset: 0x00020294

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	RdBuff	RW 0x0	Read buffer assignment per unit. When a bit is set to 0, the corresponding unit is assigned to receive read data from Read buffer 0. When set to 1, the corresponding unit is assigned to receive read data from Read buffer 1. Bit[0] corresponds to unit ID 0x0, bit[1] corresponds to unit ID 0x1, etc. For example, setting to 0x80 stands for unit ID 0x7 receives read data from read buffer 1, while the rest of the units receive data from read buffer 0.

Table 174: CIB Parity Error Status Register
Offset: 0x00020298

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0x0	Reserved
3	Port1 Read Internal RAM Error	RO 0x0	Internal error over Port 1 read RAM due to parity error.
2	Port0 Read Internal RAM Error	RO 0x0	Internal error over Port 0 read RAM due to parity error.
1	Port1 Write Internal RAM Error	RO 0x0	Internal error over Port 1 write RAM due to parity error.
0	Port0 Write Internal RAM Error	RO 0x0	Internal error over Port 0 write RAM due to parity error.

A.2.1.4 SoC Global Timers

Table 175: SoC Global Timers Control Register
Offset: 0x00020300

Bit	Field	Type/InitVal	Description
31	Reserved	RSVD 0x0	Reserved
30:28	Global Timer3 Ratio	RW 0x0	Global Timer 3 Ratio See definition and encoding at <GlobalWDTimerRatio>
27:25	Global Timer2 Ratio	RW 0x0	Global Timer 2 Ratio See definition and encoding at <GlobalWDTimerRatio>
24:22	Global Timer1 Ratio	RW 0x0	Global Timer 1 Ratio See definition and encoding at <GlobalWDTimerRatio>
21:19	Global Timer0 Ratio	RW 0x0	Global Timer 0 Ratio See definition and encoding at <GlobalWDTimerRatio>

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 175: SoC Global Timers Control Register (Continued)
Offset: 0x00020300

Bit	Field	Type/InitVal	Description
18:16	GlobalWDTimerRatio	RW 0x0	Global Watchdog Timer Ratio The field defines the Global Watchdog Timer interval. Notate the number in this field as R. When activated, the timer will be incremented upon every (2 ^ R) source clocks. Source clock frequency is L2 frequency, divided by2. When the <GlobalWDTimer25MHz> field is enabled, this field is ignored. 0 = 1: Timer tic occurs every source clock. 1 = 2: Timer tic occurs every 2 source clocks. 2 = 4: Timer tic occurs every 4 source clocks. 3 = 8: Timer tic occurs every 8 source clocks. 4 = 16: Timer tic occurs every 16 source clocks. 5 = 32: Timer tic occurs every 32 source clocks. 6 = 64: Timer tic occurs every 64 source clocks. 7 = 128: Timer tic occurs every 128 source clocks.
15	Reserved	RSVD 0x0	Reserved
14	GlobalTimer3 25MHzEn	RW 0x0	Global Timer3 25Mhz frequency Enable See description at <GlobalWDTimer25MhzEn> 0 = False 1 = True
13	GlobalTimer2 25MHzEn	RW 0x0	Global Timer2 25Mhz frequency Enable See description at <GlobalWDTimer25MhzEn> 0 = False 1 = True
12	GlobalTimer1 25MHzEn	RW 0x0	Global Timer1 25Mhz frequency Enable See description at <GlobalWDTimer25MhzEn> 0 = False 1 = True
11	GlobalTimer0 25MHzEn	RW 0x0	Global Timer0 25Mhz frequency Enable See description at <GlobalWDTimer25MhzEn> 0 = False 1 = True
10	GlobalWDTimer25MhzEn	RW 0x0	Global Watchdog Timer 25 MHz Frequency Enable Defines the count frequency of the global watchdog timer. 0 = False: Global WD timer count frequency is defined by the source clock according to the <GlobalWDTimerRatio> field. Source clock frequency is L2 frequency, divided by2 1 = True: Global WD timer count frequency is 25 MHz.
9	GlobalWDTimerAuto	RW 0x0	Global Watchdog Timer Auto Mode When this bit is clear to 0 and the <GlobalWDTimer> field in the CPU Watchdog Timer Register has reached 0, the <GlobalWDTimer> stops counting. When this bit is set to 1 and the <GlobalWDTimer> field has reached 0, the value of the <GlobalTimer0Rel> field in the GlobalTimer0 Reload Register is reloaded to the <GlobalWDTimer> field. It then continues to count.

Copyright © 2015 Marvell

Table 175: SoC Global Timers Control Register (Continued)
Offset: 0x00020300

Bit	Field	Type/InitVal	Description
8	GlobalWDTimerEn	RW 0x0	Global Watchdog Timer Enable 0 = False 1 = True
7	GlobalTimer3Auto	RW 0x0	Global Timer 3 Auto Mode When this bit is clear to 0 and the <GlobalTimer3> field in the Global Timer2 Register has reached 0, the <GlobalTimer3> stops counting. When this bit is set to 1 and the <GlobalTimer3> field has reached 0, the value of the <GlobalTimer3Rel> field in the Global Timer3 Reload Register is reloaded to the <GlobalTimer3> field. It then continues to count.
6	GlobalTimer3En	RW 0x0	CPU Timer 3 Enable 0 = False 1 = True
5	GlobalTimer2Auto	RW 0x0	Global Timer 2 Auto Mode When this bit is clear to 0 and the <GlobalTimer2> field in the Global Timer2 Register has reached to 0, the <GlobalTimer2> stops counting. When this bit is set to 1 and <GlobalTimer2> has reached zero, the value in the <GlobalTimer2Rel> field in the Global Timer1 Reload Register is reloaded to the <GlobalTimer2> field. It then continues to count.
4	GlobalTimer2En	RW 0x0	CPU Timer 2 Enable 0 = False 1 = True
3	GlobalTimer1Auto	RW 0x0	Global Timer 1 Auto Mode When this bit is clear to 0 and the <GlobalTimer1> field in the Global Timer1 Register has reached to 0, the <GlobalTimer1> stops counting. When this bit is set to 1 and the <GlobalTimer1> has reached zero, the value in the <GlobalTimer1Rel> field in the Global Timer2 Reload Register is reloaded to the <GlobalTimer1> field. It then continues to count.
2	GlobalTimer1En	RW 0x0	CPU Timer 1 Enable 0 = False 1 = True
1	GlobalTimer0Auto	RW 0x0	Global Timer 0 Auto Mode When this bit is cleared to 0 and the <GlobalTimer0> field in the Global Timer0 Register has reached 0, the <GlobalTimer0> stops counting. When this bit is set to 1 and <GlobalTimer0> has reached 0, the value in the <GlobalTimer0Rel> field in the Global Timer0 Reload Register is reloaded to the <GlobalTimer0> field. It then continues to count.
0	GlobalTimer0En	RW 0x0	Global Timer 0 Enable 0 = False 1 = True

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 176: SoC Global Timers Events Status register Register
Offset: 0x00020304

Bit	Field	Type/InitVal	Description
31	WD timer expired	RW0C 0x0	Sticky indication for watchdog timer expiration.
30:25	Reserved	RSVD 0x0	Reserved
24	Timer3 expired	RW0C 0x0	Sticky indication for timer 3 expiration.
23:17	Reserved	RSVD 0x0	Reserved
16	Timer2 expired	RW0C 0x0	Sticky indication for timer 2 expiration.
15:9	Reserved	RSVD 0x0	Reserved
8	Timer1 expired	RW0C 0x0	Sticky indication for timer 1 expiration.
7:1	Reserved	RSVD 0x0	Reserved
0	Timer0 expired	RW0C 0x0	Sticky indication for timer 0 expiration.

Table 177: SoC Global Timer 0 Reload Register
Offset: 0x00020310

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer0Rel	RW 0x0	Global Timer 0 Reload This field contains the reload value of timer 0. It is used as the reload value in Periodic mode.

Table 178: SoC Global Timer 0 Register
Offset: 0x00020314

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer0	RW 0x0	Global Timer 0 This 32-bit counter is decremented according to the the <GlobalTimer0Ratio> field. When the <GlobalTimer0En> field in the Global Timer Control Register = 0, the <GlobalTimer0> stops. When the <GlobalTimer0En> field = 1 and the <GlobalTimer0Auto> field in the Global Timer Control Register = 0, the <GlobalTimer0> is decremented until it reaches to 0. Then, it stops. When field <GlobalTimer0En> field = 1 and the <GlobalTimer0Auto> field = 1, the <GlobalTimer0> is decremented until it reaches to 0. Then the value in the <GlobalTimer0Rel> field in the Global 0 Reload Register is reload to the <GlobalTimer0>. Then, the <GlobalTimer0> continues to count. When <GlobalTimer0> reaches 0, the <GlobalTimer0IntReq> field in the Coherency Fabric Local Cause Register is set to 1.

Table 179: SoC Global Timer 1 Reload Register
Offset: 0x00020318

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer1Rel	RW 0x0	Global Timer 1 Reload See the Global Timer0 Reload Register.

Table 180: SoC Global Timer 1 Register
Offset: 0x0002031C

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer1	RW 0x0	Global Timer 1 See the Global Timer 0 Register.

Table 181: SoC Global Timer 2 Reload Register
Offset: 0x00020320

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer2Rel	RW 0x0	Global Timer 2 Reload See the Global Timer 0 Reload Register

Copyright © 2015 Marvell

Table 182: SoC Global Timer 2 Register
Offset: 0x00020324

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer2	RW 0x0	Global Timer 2 See the Global Timer 0 Register

Table 183: SoC Global Timer 3 Reload Register
Offset: 0x00020328

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer3Rel	RW 0x0	Global Timer 3 Reload See the Global Timer0 Reload Register.

Table 184: SoC Global Timer 3 Register
Offset: 0x0002032C

Bit	Field	Type/InitVal	Description
31:0	GlobalTimer3	RW 0x0	Global Timer 3 See the Global Timer 0 Register.

A.2.1.5 Shared Inbound Doorbell

Table 185: Shared Inbound Doorbell0 Register
Offset: 0x00020400

Bit	Field	Type/InitVal	Description
31:0	InboundIntCs0	RW 0x0	Inbound Command Cause When this field is not zero, the <SharedInboundDoorbellSum> field is set in the Main Interrupt Cause Register. Setting any cause bit is performed using the following two mechanisms: - An Mbus Master direct write of 1 sets the bits in this field. An Mbus Master write of 0 has no effect. - Any write to the Software Triggered Interrupt Register, targeting current Inbound Doorbell <Doorbell Sel>, sets the associated cause bit <IntID>. Any CPU write of 0 clear the bits in this field. A CPU write of 1 has no effect.

Table 186: Shared Inbound Doorbell1 Register
Offset: 0x00020404

Bit	Field	Type/InitVal	Description
31:0	InboundIntCs1	RW 0x0	See the Shared Inbound Doorbell0 Register.

Table 187: Shared Inbound Doorbell2 Register
Offset: 0x00020408

Bit	Field	Type/InitVal	Description
31:0	InboundIntCs2	RW 0x0	See the Shared Inbound Doorbell0 Register.

A.2.1.6 SoC Attribute Map

Table 188: Mbus Units Priority Control Register
Offset: 0x00020420

Bit	Field	Type/InitVal	Description
31:30	Unit 15 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=15. 0 = Low 1 = Med 2 = High 3 = Top
29:28	Unit 14 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=14. 0 = Low 1 = Med 2 = High 3 = Top
27:26	Unit 13 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=13. 0 = Low 1 = Med 2 = High 3 = Top
25:24	Unit 12 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=12. 0 = Low 1 = Med 2 = High 3 = Top
23:22	Unit 11 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=11. 0 = Low 1 = Med 2 = High 3 = Top
21:20	Unit 10 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=10. 0 = Low 1 = Med 2 = High 3 = Top
19:18	Unit 9 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=9. 0 = Low 1 = Med 2 = High 3 = Top

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 188: Mbus Units Priority Control Register (Continued)
Offset: 0x00020420

Bit	Field	Type/InitVal	Description
17:16	Unit 8 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=8. 0 = Low 1 = Med 2 = High 3 = Top
15:14	Unit 7 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=7. 0 = Low 1 = Med 2 = High 3 = Top
13:12	Unit 6 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=6. 0 = Low 1 = Med 2 = High 3 = Top
11:10	Unit 5 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=5. 0 = Low 1 = Med 2 = High 3 = Top
9:8	Unit 4 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=4. 0 = Low 1 = Med 2 = High 3 = Top
7:6	Unit 3 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=3. 0 = Low 1 = Med 2 = High 3 = Top
5:4	Unit 2 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=2. 0 = Low 1 = Med 2 = High 3 = Top
3:2	Unit 1 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=1. 0 = Low 1 = Med 2 = High 3 = Top
1:0	Unit 0 Prio	RW 0x0	Defines the priority of requests initialized by unit with source_id=0. 0 = Low 1 = Med 2 = High 3 = Top

Copyright © 2015 Marvell

Table 189: Fabric Units Priority Control Register
Offset: 0x00020424

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x3000F	Reserved

Table 190: Mbus Units Prefetch Control Register
Offset: 0x00020428

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	Unit15PFen	RW 0x0	Mbus Unit15 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit15. 0 = False 1 = True
14	Unit14PFen	RW 0x0	Mbus Unit14 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit14. 0 = False 1 = True
13	Unit13PFen	RW 0x0	Mbus Unit13 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit13. 0 = False 1 = True
12	Unit12PFen	RW 0x0	Mbus Unit12 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit12. 0 = False 1 = True
11	Unit11PFen	RW 0x0	Mbus Unit11 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit11. 0 = False 1 = True
10	Unit10PFen	RW 0x0	Mbus Unit10 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit10. 0 = False 1 = True

Copyright © 2015 Marvell

Table 190: Mbus Units Prefetch Control Register (Continued)
Offset: 0x00020428

Bit	Field	Type/InitVal	Description
9	Unit9PFen	RW 0x0	Mbus Unit9 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit9. 0 = False 1 = True
8	Unit8PFen	RW 0x0	Mbus Unit8 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit8. 0 = False 1 = True
7	Unit7PFen	RW 0x0	Mbus Unit7 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit7. 0 = False 1 = True
6	Unit6PFen	RW 0x0	Mbus Unit6 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit6. 0 = False 1 = True
5	Unit5PFen	RW 0x0	Mbus Unit5 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit5. 0 = False 1 = True
4	Unit4PFen	RW 0x0	Mbus Unit4 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit4. 0 = False 1 = True
3	Unit3PFen	RW 0x0	Mbus Unit3 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit3. 0 = False 1 = True
2	Unit2PFen	RW 0x0	Mbus Unit2 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit2. 0 = False 1 = True

Copyright © 2015 Marvell

Table 190: Mbus Units Prefetch Control Register (Continued)
Offset: 0x00020428

Bit	Field	Type/InitVal	Description
1	Unit1PFen	RW 0x0	Mbus Unit1 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit1. 0 = False 1 = True
0	Unit0PFen	RW 0x0	Mbus Unit0 Prefetch Enable Defines whether the intermediate engines should perform a prefetch action as a result of a transaction issued by Mbus Unit0. 0 = False 1 = True

A.2.1.7 MMU Static Windows

Table 191: Memory Management Unit (MMU) Static WindowN Register (N=0–7)
Offset: Windows0: 0x00020890, Windows1: 0x00020894, Windows2: 0x00020898, Windows3: 0x0002089C, Windows4: 0x000208A0, Windows5: 0x000208A4, Windows6: 0x000208A8, Windows7: 0x000208AC
Offset Formula: 0x00020890+4*N: where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
Static MMU windows.			
31:0	Reserved	RSVD 0x0	Reserved

A.2.1.8 Semaphores

Table 192: Semaphore0 Register
Offset: 0x00020500

Bit	Field	Type/InitVal	Description
31:27	Reserved	RO 0x0	Reserved
26:24	Semaphore 3	RO 0x0	Semaphore 3 See the description in the <Semaphore 0> field.
23:19	Reserved	RO 0x0	Reserved
18:16	Semaphore 2	RO 0x0	Semaphore 2 See the description in the <Semaphore 0> field.
15:11	Reserved	RO 0x0	Reserved

Document Classification: Proprietary Information

Table 192: Semaphore0 Register (Continued)
Offset: 0x00020500

Bit	Field	Type/InitVal	Description
10:8	Semaphore 1	RO 0x0	Semaphore 1 See the description in the <Semaphore 0> field.
7:3	Reserved	RO 0x0	Reserved
2:0	Semaphore 0	RO 0x0	Semaphore 0 A semphore can be locked by a CPU or an external I/O device. The first one to read it before it is locked, receives its ID value - the CPU ID or 0x4 for an external I/O device. Once locked by one of the agents, a read by any of the other agents will return the owner ID. Unlock the semaphore by writing a value of 0xFF. A write of any other value is ignored.

Table 193: Semaphoren Register (n=1–7)
Offset: Semaphore1: 0x00020504, Semaphore2: 0x00020508, Semaphore3: 0x0002050C, Semaphore4: 0x00020510, Semaphore5: 0x00020514, Semaphore6: 0x00020518, Semaphore7: 0x0002051C
Offset Formula: 0x00020504+(n - 1)*4 : where n (1-7) represents Semaphore

Bit	Field	Type/InitVal	Description
Semaphore%<n> group containing four independent semaphores.			
31:27	Reserved	RO 0x0	Reserved
26:24	Semaphore 3	RO 0x0	See the description in the Semaphor0 register.
23:19	Reserved	RO 0x0	Reserved
18:16	Semaphore 2	RO 0x0	See the description in the Semaphor0 register.
15:11	Reserved	RO 0x0	Reserved
10:8	Semaphore 1	RO 0x0	See the description in the Semaphor0 register.
7:3	Reserved	RO 0x0	Reserved
2:0	Semaphore 0	RO 0x0	See the description in the Semaphor0 register.

Copyright © 2015 Marvell

Table 194: Global Scratch Pad i Register (i=0–7)
Offset: pad id0: 0x00020520, pad id1: 0x00020540, pad id2: 0x00020560, pad id3: 0x00020580, pad id4: 0x000205A0, pad id5: 0x000205C0, pad id6: 0x000205E0, pad id7: 0x00020600
Offset Formula: 0x00020520+32*i: where i (0-7) represents pad id

Bit	Field	Type/InitVal	Description
31:0	User Defined Data	RW 0x0	User Defined Data

A.2.1.9 Clock and Reset Control

Table 195: Watchdog (WD) RSTOUTn Mask Register
Offset: 0x00020704

Bit	Field	Type/InitVal	Description
31:20	Reserved	RSVD 0x0	Reserved
19:16	CPU WD Group	RW 0x0	This field defines the CPUs that participate in the watchdog group. For each CPU<i>, when bit[i] in this field is set, the CPU is in the WD group. When the WD group is not empty (i.e. this field is NOT 0x0), at the event that all WD timers of all CPUs in the group have expired, a WD Reset will be triggered. Notice that in addition to this mechanism, Every CPU has its own WD Reset Mask that when cleared , enables it to trigger the WD Reset when its own timer has expired.
15:9	Reserved	RSVD 0x0	Reserved
8	Global WD Mask	RW 0x0	Global Watchdog (WD) Mask When this bit is set, the CPU subsystem asserts WD RSTOUTn upon global WD timer expiration.
7:2	Reserved	RSVD 0x0	Reserved
1	CPU1 WD Mask	RW 0x0	CPU1 Watchdog (WD) Mask When this bit is set, the CPU subsystem asserts WD RSTOUTn upon CPU1 WD timer expiration.
0	CPU0 WD Mask	RW 0x0	CPU0 Watchdog (WD) Mask When this bit is set, the CPU subsystem asserts WD RSTOUTn upon CPU0 WD timer expiration.

Table 196: Watchdog (WD) Interrupt Mask Register
Offset: 0x00020708

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 196: Watchdog (WD) Interrupt Mask Register (Continued)
Offset: 0x00020708

Bit	Field	Type/InitVal	Description
7:4	CPU1 Local WD Interrupt Mask	RW 0x2	CPU1 WD Interrupt Mask When bit[i] of this field is set, a CPU[i] WD timer expiration will assert the ARM_WD_INTR of CPU1.
3:0	CPU0 Local WD Interrupt Mask	RW 0x1	CPU0 WD Interrupt Mask When bit[i] of this field is set, a CPU[i] WD timer expiration will assert the ARM_WD_INTR of CPU0.

Table 197: CPUn Software Reset Control Register (n=0–1)
Offset: CPU ID0: 0x00020800, CPU ID1: 0x00020808
Offset Formula: 0x00020800+8*n : where n (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x80	Reserved
0	CPU SW RESET	Special <%n>	When this bit is set to 1, CPU will be held at reset. The CPU components held at reset are the CPU core and CPU debug interface. For CPU1: The default value will be 0x0. CPU1 is held until its bit is configured differently. For CPU0: The default value will be 0x1. CPU0 wakes and start working. 0 = Disable: CPU is not held at reset. 1 = Enable: CPU is held at reset.

A.2.1.10 Endpoint Interrupt Controller

Table 198: Endpoint Interrupt Cause Vec0 Register
Offset: 0x00020900

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28:0	ep_int_cause_0	RO 0x0	Interrupt Line n (n=0..28)

Table 199: Endpoint Interrupt Cause Vec1 Register
Offset: 0x00020904

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 199: Endpoint Interrupt Cause Vec1 Register (Continued)
Offset: 0x00020904

Bit	Field	Type/InitVal	Description
28:0	INT[n]	RO 0x0	Interrupt line n (n=29..57)

Table 200: Endpoint Interrupt Cause Vec2 Register
Offset: 0x00020908

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28:0	INT[n]	RO 0x0	Interrupt Line n (n=58..86)

Table 201: Endpoint Interrupt Cause Vec3 Register
Offset: 0x0002090C

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28:0	INT[n]	RO 0x0	Interrupt Line n (n=87..115)

Table 202: Endpoint Select Cause Register
Offset: 0x00020928

Bit	Field	Type/InitVal	Description
31	Stat	RO 0x0	Indicates if there are valid interrupts in one of the cause vectors. An interrupt becomes a valid when its cause and mask bits are set. 0 = No valid interrupts: No active non-masked interrupts in all Interrupt Cause registers. 1 = There are active interrupts: There are active non-masked in some of the Interrupt Cause registers.
30:29	VecSel	RO 0x3	Vector Select Indicates if bits[28:0] are a shadow Main Cause Vector 0, Vector 1, Vector 2 or Vector 3 after masking. 0 = Vector 0: Shadow of Cause register Vector 0 1 = Vector 1: Shadow of Cause register Vector 1 2 = Vector 2: Shadow of Cause register Vector 2 3 = Vector 3: Shadow of Cause register Vector 3

Copyright © 2015 Marvell

Table 202: Endpoint Select Cause Register (Continued)
Offset: 0x00020928

Bit	Field	Type/InitVal	Description
28:0	Cause	RO 0x0	A shadow register of the Main Cause Vector 0, Vector 1, Vector 2 or Vector 3 after the masking operation (valid interrupts) Return the Endpoint masked Cause Register Vector 0, if any interrupt is still set after masking in this vector, else return masked Cause Register Vector 1, if any interrupt is still set after masking in this vector, else return masked Cause Register Vector 2 if any interrupt is still set after masking in this vector, else return masked Cause Register Vector 3.

Table 203: Endpoint Fabric Local Interrupt Mask Register
Offset: 0x00020940

Bit	Field	Type/InitVal	Description
31:0	EPFabricLocaMask	RW 0x0	Endpoint Coherency Fabric Local Interrupt Mask Mask bit per Coherency Fabric Local Interrupt event. If a bit is set to 0, the interrupt is masked; if set to 1 the interrupt is enabled. The mask only affects the assertion of the associated interrupt line to the Endpoint main cause register.

Table 204: Endpoint SoC Main Interrupt Error 0 Mask Register
Offset: 0x00020944

Bit	Field	Type/InitVal	Description
31:0	EPSocErrorMask	RW 0x0	SoC Error Interrupt Mask. Mask bit per SoC Main Interrupt Error event. If the bit is set to 0, the interrupt is masked. If it is set to 1, the interrupt is enabled. The mask only affects the assertion of the SoC Error Sum interrupt line Endpoint main interrupt cause registers.

Table 205: Endpoint SoC Main Interrupt Error 1 Mask Register
Offset: 0x00020948

Bit	Field	Type/InitVal	Description
31:0	EPSocErrorMask1	RW 0x0	SoC Error Interrupt Mask for [63:32] additional error events. Mask bit per SoC Main Interrupt Error event. If the bit is set to 0, the interrupt is masked. If it is set to 1, the interrupt is enabled. The mask only affects the assertion of the SoC Error Sum interrupt line Endpoint main interrupt cause registers.

Copyright © 2015 Marvell

A.2.1.11 Main Interrupt Controller

Table 206: Software Triggered Interrupt Register
Offset: 0x00020A04

Bit	Field	Type/InitVal	Description
The Register controls the issuing of software interrupts.			
31:26	Reserved	RSVD 0x0	Reserved
25:24	Target List Filter	WO 0x0	Target List Filter Controls the filtering that the distributor performs on software interrupts as follows: 0 = All and Self: Send the interrupt to all CPUs that are defined to be in the Target List. 1 = All Not Self: Send the interrupt to all CPUs that are defined to be in the Target List except the CPU that issues the request. 2 = Only Self: Send the interrupt only to the CPU interface that requested the interrupt. 3 = Reserved
23:10	Reserved	RSVD 0x0	Reserved
9	CPU1 in Target List	WO 0x0	CPU1 In Interrupt Target List When this bit is set, the distributor includes CPU1 in Target List and signals an interrupt accordingly. For the usage of the Target List, see the description of the <Target List Filter> field.
8	CPU0 in Target List	WO 0x0	CPU0 In Interrupt Target List When this bit is set, the distributor includes CPU0 in Target List and signals an interrupt accordingly. For the usage of the Target List, see the description of the <Target List Filter> field.
7	Reserved	RSVD 0x0	Reserved
6:5	Inbound Doorbell Sel	WO 0x0	Inbound Doorbell Select Specifies on which Inbound Doorbell register the interrupt specified in the IntID should be set. 0 = Doorbell0: Inbound Doorbell 0 register, per-CPU register 1 = Doorbell1: Shared Inbound Doorbell 1 register 2 = Doorbell2: Shared Inbound Doorbell 2 register 3 = Doorbell3: Shared Inbound Doorbell 3 register
4:0	IntID	WO 0x0	Set this to the INTID of the software interrupt that the distributor signals to the processor. The INITID is latched in the CPU Inbound Doorbell Cause register.

Copyright © 2015 Marvell

Table 207: SoC Main Interrupt Error 0 Cause Register
Offset: 0x00020A20

Bit	Field	Type/InitVal	Description
31:0	Err0 Int[n]	RO 0x0	Error Interrupt Line n where n=0..31. The associated mask register is defined for CPU.

Table 208: SoC Main Interrupt Error 1 Cause Register
Offset: 0x00020A24

Bit	Field	Type/InitVal	Description
31:0	Err1 Int[n]	RO 0x0	Error Interrupt Line n where n=32..63. The associated mask register is defined for CPU.

Table 209: Endpoint ISE Register
Offset: 0x00020A30

Bit	Field	Type/InitVal	Description
Endpoint Interrupt Set Enable (ISE)			
31:10	Reserved	RSVD 0x0	Reserved
9:0	Interrupt ID	WO 0x0	Writing the Interrupt ID to this field, sets the IntEN bit in the Interrupt Source Control register that is associated with the Interrupt ID. This means that interrupt delivery is now enabled.

Table 210: Endpoint ICE Register
Offset: 0x00020A34

Bit	Field	Type/InitVal	Description
Endpoint Interrupt Clear Enable (ICE)			
31:10	Reserved	RSVD 0x0	Reserved
9:0	Interrupt ID	WO 0x0	Writing the Interrupt ID to this field, clears the IntEN bit in the Interrupt Source Control register that is associated with the Interrupt ID. This means that interrupt delivery is now disabled.

Copyright © 2015 Marvell

Table 211: Interrupt Source i Control Register (i=0–28)
Offset: Interrupt ID0: 0x00020B00, Interrupt ID1: 0x00020B04...Interrupt ID28: 0x00020B70
Offset Formula: 0x00020B00+4*i: where i (0-28) represents Interrupt ID

Bit	Field	Type/InitVal	Description
31	EP Mask<i>	RW 0x0	Endpoint Maski When this bit is cleared, interrupt i is masked on reaching the PCIe Endpoint (EP).
30:29	Reserved	RSVD 0x0	Reserved
28	Reserved	RO 0x1	Reserved Must be 0x1.
27:24	Priority<i>	RW 0x1	Interrupt Priority Specifies the interrupt priority. The lowest priority is 0 and the highest priority is 15. A priority level of 0 disables interrupts from this source.
23:18	Reserved	RSVD 0x0	Reserved
17	Endpoint Seli CPU1	RW 0x0	Interrupt Cause Selector for CPU1 for PCIe EP
16	Endpoint Seli CPU0	RW 0x0	Interrupt Cause Selector for CPU0 for PCIe EP
15:10	Reserved	RSVD 0x0	Reserved
9	FIQ Maski CPU1	RW 0x0	CPU1 FIQ Mask Bit Interrupt i See the description for FIQ Maski CPU0.
8	FIQ Maski %CPU0	RW 0x0	CPU0 FIQ Mask Bit Interrupt i When this field is cleared, interrupt i is masked to the CPU. When this field is set, interrupt i is enabled to the CPU. The mask affects the assertion of the FIQ pin of the CPU. It does not affect the setting of bits in the cause register. The associated CPU can update this field by writing to its private IMS and ICM registers as follows: Writing i to Interrupt_ID field and 0x1 to selFIQ in the ISM sets this field. Writing i to Interrupt_ID field and 0x1 to selFIQ in the ICM clears this field. **NOTE:** This mechanism provides the support for a CPU to handle FIQ interrupts without CPU identification relativity.
7:2	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 211: Interrupt Source i Control Register (i=0–28) (Continued)
Offset: Interrupt ID0: 0x00020B00, Interrupt ID1: 0x00020B04...Interrupt ID28: 0x00020B70
Offset Formula: 0x00020B00+4*i: where i (0-28) represents Interrupt ID

Bit	Field	Type/InitVal	Description
1	IRQ Maski CPU1	RW 0x0	CPU1 IRQ Mask Bit Interrupt i See the description for IRQ Maski CPU0.
0	IRQ Maski %CPU0	RW 0x0	CPU0 IRQ Mask Bit Interrupt i When this field is cleared, interrupt i is masked to the CPU. When this field is set, interrupt i is enabled to the CPU. The mask affects the assertion of IRQ pin of the CPU. It does not affect the setting of bits in the cause register. The associated CPU can update this field by writing to its private IMS and ICM registers as follows: Writing i to Interrupt_ID field and 0x0 to selFIQ in the ISM sets this field. Writing i to Interrupt_ID field and 0x0 to selFIQ in the ICM clears this field. Notice that this mechanism provides the support for a CPU to handle IRQ interrupts without CPU identification relativity.

Table 212: Endpoint Source i Control Register (i=29–115)
Offset: Interrupt ID29: 0x00020B74, Interrupt ID30: 0x00020B78...Interrupt ID115: 0x00020CCC
Offset Formula: 0x00020B00+4*i: where i (29-115) represents Interrupt ID

Bit	Field	Type/InitVal	Description
31	EP Mask<i>	RW 0x0	Endpoint Maski When this bit is cleared, interrupt i is masked on reaching PCIe EP.
30:0	Reserved	RSVD 0x1000000	Reserved

A.2.1.12 CoreSight Control

Table 213: CoreSight Components Base Address Register
Offset: 0x00020D00

Bit	Field	Type/InitVal	Description
31:12	CoreSightComponentBAR	RW 0x42300	CoreSight Components Base Addresses as registered within ROM Table. The register initial value points to the ROM Table Base Address. List of CoreSight component BARs per CPU and over the SoC: 0x4230_0000ROM Table 0x4231_0000CPU0 Debug 0x4231_1000CPU0 PMU 0x4231_8000CPU0 CTI 0x4231_C000CPU0 PTM 0x4232_4000SoC ETB0 0x4232_5000SoC CTI (ETB0 + ETB1 triggers) 0x4231_2000CPU1 Debug 0x4231_3000CPU1 PMU 0x4231_9000CPU1 CTI 0x4231_D000CPU1 PTM 0x4232_6000SoC ETB1 0x4232_0000SoC Timestamp
11:0	Reserved	RSVD 0x0	Reserved

Table 214: CoreSight Status Register
Offset: 0x00020D0C

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

A.2.1.13 CA9 MP Core SoC Control

Table 215: CA9 MP Core Reset Control Register
Offset: 0x00020D84

Bit	Field	Type/InitVal	Description
31:19	Reserved	RSVD 0x1FFF	Reserved
18:17	WD Reset	RW 0x3	Watchdog (WD) reset async Flip-Flop (FF) reset Acive Low (AL). 1 bit per CPU: Bit [17] sets CPU0, and bit[18] sets CPU1. WD reset request from CPU is a pulse, later translated to level signal to the SoC. To de-assert the WD reset level signal to the SoC for each CPU, set its bit to 0x0. Then, set its bit to 0x1 again.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 215: CA9 MP Core Reset Control Register (Continued)
Offset: 0x00020D84

Bit	Field	Type/InitVal	Description
16	Mask CS Debug Reset	RW 0x1	Mask CoreSight Debug Reset Use this bit to mask APB debug reset to Sbay/CoreSight, when both CPU-0 and CPU-1 are at reset. The default of this bit is 1. Use this bit to select whether or not to reset the APB debug when CPUs are at reset.
15:4	Reserved	RSVD 0x0	Reserved
3	MC Hint Mode En	RW 0x0	Memory Controller (MC) Hint Mode Enable 0 = Legacy mode: CIB priority and prefetch transactions determined by static values per Mbus source ID. 1 = Hint mode: CIB priority and prefetch transactions determined with CBE [19:18], [15] accordingly per transaction.
2:0	Reserved	RSVD 0x0	Reserved

A.2.1.14 CPU Local

Table 216: CPU Configuration Register (N=0–1)
Offset: CPU ID0: 0x00021800, CPU ID1: 0x00021900
Offset Formula: 0x00021800+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:11	Reserved	RSVD 0x205A0	Reserved
10:9	Core Mode	RW SAR	Specifies the targeted ARM core behavior 1 = Cortex-A8: (v7, uP) 3 = Cortex-A9: (v7, uP or MP)
8:7	Reserved	RSVD 0x0	Reserved
6	NMFI En	RW SAR	Fast Interrupt Enable Configures fast interrupts to be nonmaskable. This pin is only sampled during reset of the processor. 0 = NMFI Disable: Clears the NMFI bit in the CP15 System Control Register 1 = NMFI Enable: Sets the NMFI bit in the CP15 System Control Register

Copyright © 2015 Marvell

Table 216: CPU Configuration Register (N=0–1) (Continued)
Offset: CPU ID0: 0x00021800, CPU ID1: 0x00021900
Offset Formula: 0x00021800+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
5	TE Init	RW SAR	Thumb Exception Initialization Controls the state of the TE bit in the CP15 System Control Register. This pin is only sampled during reset of the processor. 0 = ARM Exception: Enable ARM exception generation. On exception entry, CPSR T bit is 0 and J bit is 0. 1 = Thumb Exception: Enable Thumb exception generation. On exception entry, CPSR T bit is 1 and J bit is 0.
4	Reserved	RW 0x1	Reserved Must be 0x1.
3	Endian Init	RW SAR	Endian Initialization Controls the state of the B bit in the CP15 c1 Control register. This pin is only sampled during reset of the processor. 0 = Little Endian: Clears the B bit in the CP15 c1 Control Register, indicating Little Endian memory system support. 1 = Big Endian: Sets the B bit in the CP15 c1 Control Register, indicating Big Endian word-invariant memory system support.
2	Reserved	RSVD 0x0	Reserved
1	Vec Init Loc	RW SAR	Determines the reset location of the boot starting address. 0 = MsbZeros: Boot starting address is 0x00000000 1 = MsbOnes: Boot starting address is 0xFFFF0000.
0	Reserved	RSVD 0x0	Reserved

Table 217: CPU Control and Status Register (N=0–1)
Offset: CPU ID0: 0x00021808, CPU ID1: 0x00021908
Offset Formula: 0x00021808+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x8000	Reserved
1	CPU Big Endian Status	RO 0x0	Big Endian Status Reflects the value of the Big Endian field of the CPU CP15 register. 0 = Little 1 = Big

Copyright © 2015 Marvell

Table 217: CPU Control and Status Register (N=0–1) (Continued)
Offset: CPU ID0: 0x00021808, CPU ID1: 0x00021908
Offset Formula: 0x00021808+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
0	Reserved	RSVD 0x0	Reserved

Table 218: CPU IO Sync Barrier Control Register (N=0–1)
Offset: CPU ID0: 0x00021810, CPU ID1: 0x00021910
Offset Formula: 0x00021810+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:1	Reserved	RO 0x0	Reserved
0	Trigger And Status	RW 0x0	Sync Barrier Trigger and Status Triggers sync barrier operation for the CPU core. 0 = IDLE 1 = BUSY

Table 219: SoC Private Timers Control Register (N=0–1)
Offset: CPU ID0: 0x00021840, CPU ID1: 0x00021940
Offset Formula: 0x00021840+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:25	Reserved	RSVD 0x0	Reserved
24:22	Private Timer1 Ratio	RW 0x0	Private Timer 1 Ratio The field defines the Private Timer interval. The number in this field is R in the following formula. When activated, the timer will be incremented upon every (2 ^ R) source clocks. When the <Private Timer 25MHz En> field is set, this field is ignored.
21:19	Private Timer0 Ratio	RW 0x0	Private Timer 0 Ratio The field defines the Private Timer interval. The number in this field is R in the following formula. When activated, the timer will be incremented upon every (2 ^ R) source clocks. When the <Private Timer 25MHz En> field is set, this field is ignored.
18:13	Reserved	RSVD 0x0	Reserved
12	Private Timer1 25MHz En	RW 0x0	Private Timer1 25 MHz Frequency Enable Defines the count frequency of the Private timer1. 0 = False 1 = True
11	Private Timer0 25MHz En	RW 0x0	Private Timer0 25 MHz Frequency Enable Defines the count frequency of the Private timer0. 0 = False 1 = True

Copyright © 2015 Marvell

Table 219: SoC Private Timers Control Register (N=0–1) (Continued)
Offset: CPU ID0: 0x00021840, CPU ID1: 0x00021940
Offset Formula: 0x00021840+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
10:4	Reserved	RSVD 0x0	Reserved
3	Private Timer1 Auto	RW 0x0	Private Timer 1 Auto Mode When this bit is cleared to 0 and the <PrivateTimer1> field in the Private Timer0 register has reached zero, the <PrivateTimer1> field stops counting. When this bit is set to 1 and the <PrivateTimer1> field has reached zero, the <Private Timer1 Rel> field in the Private Timer1 Reload register is reloaded to the <Private Timer1> field value. It then continues to count.
2	Private Timer1 En	RW 0x0	Private Timer 1 Enable 0 = False 1 = True
1	Private Timer0 Auto	RW 0x0	Private Timer 0 Auto Mode When this bit is cleared to 0 and the <Private Timer0> field in the Private Timer0 register has reached zero, the <Private Timer0> field stops counting. When this bit is set to 1 and the <Private Timer0> field has reached zero, the <Private Timer0 Rel> field in the Private Timer0 Reload register is reloaded to the <Private Timer0> field value. It then continues to count.
0	Private Timer0 En	RW 0x0	Private Timer 0 Enable 0 = False 1 = True

Table 220: SoC Private Timer 0 Reload Register (N=0–1)
Offset: CPU ID0: 0x00021850, CPU ID1: 0x00021950
Offset Formula: 0x00021850+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	Private Timer0 Rel	RW 0x0	Private Timer 0 Reload This field contains the reload value of private timer 0. It is used as the reload value in Periodic mode.

Copyright © 2015 Marvell

Table 221: SoC Private Timer 0 Register (N=0–1)
Offset: CPU ID0: 0x00021854, CPU ID1: 0x00021954
Offset Formula: 0x00021854+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	PrivateTimer0	RW 0x0	Private Timer 0 This 32-bit counter is decremented every device internal clock. When the <PrivateTimer0En> field in the Private Timer Control Register = 0, the <PrivateTimer0> field stops. When the <PrivateTimer0En> field = 1 and the <PrivateTimer0Auto> field in the Private Timer Control register = 0, the <PrivateTimer0> field is decremented until it reaches to 0. Then it stops. When the <PrivateTimer0En> field = 1 and the <PrivateTimer0Auto> field = 1, the <PrivateTimer0> field is decremented until it reaches to 0, then the <PrivateTimer0Rel> field in the Private 0 Reload Register is reload to the <PrivateTimer0> field and then the <PrivateTimer0> field continues to count. When the <PrivateTimer0> field reaches 0, the <PrivateTimer0IntReq> field in the Mbus-L to Mbus Bridge Interrupt Cause Register is set to 1.

Table 222: SoC Private Timer 1 Reload Register (N=0–1)
Offset: CPU ID0: 0x00021858, CPU ID1: 0x00021958
Offset Formula: 0x00021858+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	Private Timer1 Rel	RW 0x0	Private Timer 1 Reload See the Private Timer 0 Reload Register.

Table 223: SoC Private Timer 1 Register (N=0–1)
Offset: CPU ID0: 0x0002185C, CPU ID1: 0x0002195C
Offset Formula: 0x0002185C+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	Private Timer1	RW 0x0	Private Timer 1 See the Private Timer 0 register.

Table 224: SoC Private Timers Events Status Register (N=0–1)
Offset: CPU ID0: 0x00021868, CPU ID1: 0x00021968
Offset Formula: 0x00021868+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8	Private Timer1 Expired	RW0C 0x0	Indication that Private Timer 1 expired.
7:1	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 224: SoC Private Timers Events Status Register (N=0–1) (Continued)
Offset: CPU ID0: 0x00021868, CPU ID1: 0x00021968
Offset Formula: 0x00021868+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
0	Private Timer0 Expired	RW0C 0x0	Indication that Private Timer 0 expired.

Table 225: Outbound Doorbell Register (N=0–1)
Offset: CPU ID0: 0x00021870, CPU ID1: 0x00021970
Offset Formula: 0x00021870+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	OutboundIntCs	RW 0x0	Outbound Interrupt Cause When a bit in this field is set and the corresponding bit in <OutboundIntCsMask> in the Outbound Doorbell Mask Register is also set, bit <OutboundDoorbell> is set in the Main Interrupt Cause Register. A CPU write of 1 set the bits in this field. A CPU write of 0 has no effect. A target write of 0 clear the bits in this field. A target write of 1 has no affect

Table 226: Outbound Doorbell Mask Register (N=0–1)
Offset: CPU ID0: 0x00021874, CPU ID1: 0x00021974
Offset Formula: 0x00021874+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	OutboundIntCsMask	RW 0x0	Outbound Interrupt Cause Mask Mask bit per each cause bit in <OutboundIntCs> field in the Outbound Doorbell Register. The mask only affects the assertion of the interrupt bit in the Main Interrupt Cause Register. It does not affect the setting of bits in the Outbound Doorbell Register.

Copyright © 2015 Marvell

Table 227: Inbound Doorbell Register (N=0–1)
Offset: CPU ID0: 0x00021878, CPU ID1: 0x00021978
Offset Formula: 0x00021878+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	InboundIntCs	RW 0x0	Inbound Command Cause When this bit is not zero and the corresponding bit <InboundIntCsMask> in the Inbound Doorbell Mask Register is set: - The <InboundDoorbellSumLow> field is set in the Main Interrupt Cause Register for the lowest 16 doorbells. and -The <InboundDoorbellSumHigh> field is set for the upper 16 doorbells. Setting any non-masked doorbell cause bit is performed using the following 2 mechanisms: - An initiator direct write of 1 sets the bits in this field. An initiator write of 0 has no effect. - Other agent in the writes to the Software Triggered Interrupt Register, targeting current CPU with specified Msg ID A CPU write of 0 clear the bits in this field. A CPU write of 1 has no effect.

Table 228: Inbound Doorbell Mask Register (N=0–1)
Offset: CPU ID0: 0x0002187C, CPU ID1: 0x0002197C
Offset Formula: 0x0002187C+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	InboundIntCsMask	RW 0x0	Inbound Interrupt Cause Mask Mask bit per each cause bit in the <InboundIntCs> field in the Inbound Doorbell Register. The mask only affects the assertion of interrupt bit in Main Interrupt Cause Register. It does not affect the setting of bits in the Inbound Doorbell Register.

Table 229: SoC PPI Cause Reg Register (N=0–1)
Offset: CPU ID0: 0x00021880, CPU ID1: 0x00021980
Offset Formula: 0x00021880+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:29	Reserved	RO 0x0	Reserved
28:0	INT[n]	RO 0x0	Interrupt Line n (n=0..28)

Copyright © 2015 Marvell

Table 230: SoC Private Peripheral Interrupt Set Mask (PPI ISM) Register (N=0–1)
Offset: CPU ID0: 0x000218B8, CPU ID1: 0x000219B8
Offset Formula: 0x000218B8+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
SoC Private Peripheral Interrupt Set Mask			
31	Set FIQ	WO 0x0	Select Set FIQ 0 = Set IRQ Mask: Mask IRQ of the requesting CPU. 1 = Set FIQ Mask: Mask FIQ of the requesting CPU.
30:10	Reserved	RSVD 0x0	Reserved
9:0	Interrupt ID	WO 0x0	Interrupt ID Writing the Interrupt ID to this field masks the IRQ/FIQ of the accessing CPU in the Interrupt Source Control register associated with the SoC Private Peripheral Interrupts. **NOTE:** Writing a value that is greater than 31 to this field can cause unpredicable behavior.

Table 231: SoC Private Peripheral Interrupt Clear Mask (PPI ICM) Register (N=0–1)
Offset: CPU ID0: 0x000218BC, CPU ID1: 0x000219BC
Offset Formula: 0x000218BC+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31	Clear FIQ	WO 0x0	Select Clear FIQ 0 = Clear IRQ Mask: Unmasks IRQ of the requesting CPU. 1 = Clear FIQ Mask: Unmasks FIQ of the requesting CPU.
30:10	Reserved	RSVD 0x0	Reserved
9:0	Interrupt ID	WO 0x0	Interrupt ID Writing the Interrupt ID to this field unmasks the IRQ/FIQ of the accessing CPU in the Interrupt Source Control register associated with the SoC Private Peripheral Interrupts. **NOTE:** Writing a value that is greater than 31 to this field can cause unpredicable behavior.

Table 232: SoC Main Interrupt Error 0 Mask Register (N=0–1)
Offset: CPU ID0: 0x000218C0, CPU ID1: 0x000219C0
Offset Formula: 0x000218C0+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	Mask0	RW 0x0	SoC Error 0 Interrupt Mask A mask bit per event of those in the SoC Main Interrupt Error0 Cause register. This register is banked per CPU. The mask affects the assertion of the SOCErrSum interrupt line in the SoC PPI Cause register associated with the CPU.

Copyright © 2015 Marvell

Table 233: SoC Main Interrupt Error 1 Mask Register (N=0–1)
Offset: CPU ID0: 0x000218C8, CPU ID1: 0x000219C8
Offset Formula: 0x000218C8+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	Mask1	RW 0x0	SoC Error 1 Interrupt Mask A mask bit per event of those in the SoC Main Interrupt Error 1 Cause register. This register is banked per CPU. The mask affects the assertion of the SOCErrSum interrupt line in the SoC PPI Cause register associated with the CPU. 0 = Masked: Error event is masked. 1 = Enabled: Error event is enabled.

Table 234: Local Scratch Pad 0 Register (N=0–1)
Offset: CPU ID0: 0x000218D0, CPU ID1: 0x000219D0
Offset Formula: 0x000218D0+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 235: Local Scratch Pad 1 Register (N=0–1)
Offset: CPU ID0: 0x000218D4, CPU ID1: 0x000219D4
Offset Formula: 0x000218D4+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 236: Local Scratch Pad 2 Register (N=0–1)
Offset: CPU ID0: 0x000218D8, CPU ID1: 0x000219D8
Offset Formula: 0x000218D8+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 237: Local Scratch Pad 3 Register (N=0–1)
Offset: CPU ID0: 0x000218DC, CPU ID1: 0x000219DC
Offset Formula: 0x000218DC+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Copyright © 2015 Marvell

Table 238: Local Scratch Pad 4 Register (N=0–1)
Offset: CPU ID0: 0x000218E0, CPU ID1: 0x000219E0
Offset Formula: 0x000218E0+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 239: Local Scratch Pad 5 Register (N=0–1)
Offset: CPU ID0: 0x000218E4, CPU ID1: 0x000219E4
Offset Formula: 0x000218E4+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 240: Local Scratch Pad 6 Register (N=0–1)
Offset: CPU ID0: 0x000218E8, CPU ID1: 0x000219E8
Offset Formula: 0x000218E8+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 241: Local Scratch Pad 7 Register (N=0–1)
Offset: CPU ID0: 0x000218EC, CPU ID1: 0x000219EC
Offset Formula: 0x000218EC+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:0	UserDefinedData	RW 0x0	User Defined Data

Table 242: CPU Status Register (N=0–1)
Offset: CPU ID0: 0x000218F0, CPU ID1: 0x000219F0
Offset Formula: 0x000218F0+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x0	Reserved
1	CPU Privilege Mode	RO 0x0	Select User or Privileged mode of Cortex-A9 processor. 0 = User mode 1 = Privileged mode
0	CPU Secure Mode	RO 0x0	Select security status of Cortex-A9 processor. 0 = Non-secured 1 = Secured

Copyright © 2015 Marvell

Table 243: CPU External Debug Register (N=0–1)
Offset: CPU ID0: 0x000218F4, CPU ID1: 0x000219F4
Offset Formula: 0x000218F4+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x0	Reserved
2	Debug CPU Done	RO 0x0	The processor asserts the <Debug CPU Done> field only after it has completed all non-debug state memory accesses. Indicates that all memory accesses issued by the Cortex-A9 processor result from operations performed by a debugger.
1	Debug ACK	RO 0x0	External Debug request was acknowledged by the processor.
0	External Debug Req	RW 0x0	External Debug Request

Table 244: CPU Data Engine Flags Register (N=0–1)
Offset: CPU ID0: 0x000218F8, CPU ID1: 0x000219F8
Offset Formula: 0x000218F8+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:7	Reserved	RSVD 0x0	Reserved
6:0	Data Eng Flags	RO 0x0	Cortex-A9 Data Engine Output Flags Data Eng Flags[4:0]: The value of the FPSCR[4:0] processor register. Data Eng Flags[5]: The value of the FPSCR[7] processor register. Data Eng Flags[6]: The value of the FPSCR[27] processor register.

Table 245: CPU Event Control Register (N=0–1)
Offset: CPU ID0: 0x000218FC, CPU ID1: 0x000219FC
Offset Formula: 0x000218FC+0x100*N : where N (0-1) represents CPU ID

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0x0	Reserved
3	Standby WFI	RO 0x0	Indicates whether Cortex-A9 is in WFI (Wait for Interrupt) Standby mode.
2	Standby WFE	RO 0x0	Indicates whether Cortex-A9 is in WFE (Wait for Event) Standby mode.
1	Event In	RW 0x0	Event input for Cortex-A9 processor to wake-up from WFE Standby mode.
0	Event Out	RO 0x0	Event output that is active upon SEV (Send Event) instruction execution.

Copyright © 2015 Marvell

A.2.1.15 Timestamp

Table 246: Timestamp Counter Enable Register
Offset: 0x00023000

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	Timestamp Counter Enable	RW 0x0	Enables or disables 48-bit timestamp gray counter. 0 = Disable 1 = Enable

Table 247: Timestamp Counter Preload 1 Register
Offset: 0x00023004

Bit	Field	Type/InitVal	Description
31:0	Timestamp Preload Value	RW 0x0	Preloads the [31:0] bit timestamp counter.

Table 248: Timestamp Counter Preload 2 Register
Offset: 0x00023008

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	Timestamp Preload Value	RW 0x0	Preloads the [47:32] bit timestamp counter.

Table 249: Timestamp Claim Tag Set Register
Offset: 0x00023FA0

Bit	Field	Type/InitVal	Description
31:0	Claim Tag Set	RW 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 250: Timestamp Claim Tag Clear Register
Offset: 0x00023FA4

Bit	Field	Type/InitVal	Description
31:0	Claim Tag Clear	RW 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Copyright © 2015 Marvell

Table 251: Timestamp Lock Access Register
Offset: 0x00023FB0

Bit	Field	Type/InitVal	Description
31:0	Lock Access	WO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 252: Timestamp Lock Status Register
Offset: 0x00023FB4

Bit	Field	Type/InitVal	Description
31:0	Lock Status	RO 0x3	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 253: Timestamp Authentication Status Register
Offset: 0x00023FB8

Bit	Field	Type/InitVal	Description
31:0	Authentication Status	RO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 254: Timestamp Device ID Register
Offset: 0x00023FC8

Bit	Field	Type/InitVal	Description
31:0	Device ID	RO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 255: Timestamp Device Type Identifier Register
Offset: 0x00023FCC

Bit	Field	Type/InitVal	Description
31:0	Device Type Identifier	RO 0x4	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 256: Timestamp Peripheral ID4 Register
Offset: 0x00023FD0

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID4	RO 0x5	See the register details in the CoreSight Design Kit Technical Reference Manual.

Copyright © 2015 Marvell

Table 257: Timestamp Peripheral ID5 Register
Offset: 0x00023FD4

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID5	RO 0x9A	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 258: Timestamp Peripheral ID6 Register
Offset: 0x00023FD8

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID6	RO 0xE	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 259: Timestamp Peripheral ID7 Register
Offset: 0x00023FDC

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID7	RO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 260: Timestamp Peripheral ID0 Register
Offset: 0x00023FE0

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID0	RO 0x3	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 261: Timestamp Peripheral ID1 Register
Offset: 0x00023FE4

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID1	RO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 262: Timestamp Peripheral ID2 Register
Offset: 0x00023FE8

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID2	RO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Copyright © 2015 Marvell

Table 263: Timestamp Peripheral ID3 Register
Offset: 0x00023FEC

Bit	Field	Type/InitVal	Description
31:0	Peripheral ID3	RO 0x0	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 264: Timestamp Component ID0 Register
Offset: 0x00023FF0

Bit	Field	Type/InitVal	Description
31:0	Component ID0	RO 0xD	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 265: Timestamp Component ID1 Register
Offset: 0x00023FF4

Bit	Field	Type/InitVal	Description
31:0	Component ID1	RO 0x90	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 266: Timestamp Component ID2 Register
Offset: 0x00023FF8

Bit	Field	Type/InitVal	Description
31:0	Component ID2	RO 0x5	See the register details in the CoreSight Design Kit Technical Reference Manual.

Table 267: Timestamp Component ID3 Register
Offset: 0x00023FFC

Bit	Field	Type/InitVal	Description
31:0	Component ID3	RO 0xB1	See the register details in the CoreSight Design Kit Technical Reference Manual.

Copyright © 2015 Marvell

A.3 CPU Power Management Unit Registers

The following table provides a summarized list of all registers that belong to the CPU Power Management Unit, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 268: Summary Map Table for the CPU Power Management Unit Registers

Register Name	Offset	Table and Page
CPU0 Power Management		
CPU0 Statistic Counter Global Control Register	0x00022100	Table 269, p. 508
CPU0 PM Control and Configuration Register	0x00022104	Table 270, p. 509
CPU0 PM Status and Mask Register	0x0002210C	Table 271, p. 509
CPU0 Idle Timer Value Register	0x00022110	Table 272, p. 510
CPU0 Idle High Threshold Register	0x00022114	Table 273, p. 510
CPU0 Idle Low Threshold Register	0x00022118	Table 274, p. 511
CPU0 Idle Result Register	0x0002211C	Table 275, p. 511
CPU0 PM Event Status and Mask Register	0x00022120	Table 276, p. 511
CPU0 Boot Address Redirect Register	0x00022124	Table 277, p. 512
CPU0 Resume Control Register	0x00022128	Table 278, p. 512
CPU1 Power Management		
CPU1 Statistic Counter Global Control Register	0x00022200	Table 279, p. 512
CPU1 PM Control and Configuration Register	0x00022204	Table 280, p. 513
CPU1 PM Status and Mask Register	0x0002220C	Table 281, p. 513
CPU1 Idle Timer Value Register	0x00022210	Table 282, p. 514
CPU1 Idle High Threshold Register	0x00022214	Table 283, p. 514
CPU1 Idle Low Threshold Register	0x00022218	Table 284, p. 515
CPU1 Idle Result Register	0x0002221C	Table 285, p. 515
CPU1 PM Event Status and Mask Register	0x00022220	Table 286, p. 515
CPU1 Boot Address Redirect Register	0x00022224	Table 287, p. 516
CPU1 Resume Control Register	0x00022228	Table 288, p. 516

A.3.1 CPU0 Power Management

Table 269: CPU0 Statistic Counter Global Control Register
Offset: 0x00022100

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	CPU0StatCountEn	RW 0x0	CPU0 statistics counter enable control. 0 = Disable: Statistic counter disabled. 1 = Enable: Statistic counter enabled.

Copyright © 2015 Marvell

Table 270: CPU0 PM Control and Configuration Register
Offset: 0x00022104

Bit	Field	Type/InitVal	Description
31:19	Reserved	RSVD 0x0	Reserved
18	CPU0DfsReq	RW 0x0	CPU0 DFS request. Cleared by HW.
17	Reserved	RSVD 0x0	Reserved
16	CPU0PowerDownReq	RW 0x0	CPU0 power down request. Cleared by HW.
15:8	Reserved	RSVD 0x0	Reserved
7:4	CPU0FrequencyID	RW 0x1	Indicates CPU0 Frequency ID.
3:0	CPU0PowerDomainID	RW 0x1	Indicates CPU0 Power-Domain ID.

Table 271: CPU0 PM Status and Mask Register
Offset: 0x0002210C

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25	CPU0Intr1Mask	RW 0x0	Controls the masking control of the unit's wake1 indication, as an interrupt. A value of 1 means that this interrupt signals will be masked from the unit. 0 = NoMask: Do not mask the unit's interrupt. 1 = Mask: Mask the unit's interrupt.
24	CPU0Intr0Mask	RW 0x0	Controls the masking control of the unit's wake0 indication, as an interrupt. A value of 1 means that this interrupt signals will be masked from the unit. 0 = NoMask: Do not mask the unit's interrupt. 1 = Mask: Mask the unit's interrupt.
23	CPU0Wake3Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
22	CPU0Wake2Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
21	CPU0Wake1Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
20	CPU0Wake0Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
19	Reserved	RSVD 0x0	Reserved
18	CPU0Idle2Mask	RW 0x0	A value of 1 indicates that the PMU will wait for corresponding idle indication before starting the power down flow.

Copyright © 2015 Marvell

Table 271: CPU0 PM Status and Mask Register (Continued)
Offset: 0x0002210C

Bit	Field	Type/InitVal	Description
17	CPU0Idle1Mask	RW 0x0	A value of 1 indicates that the PMU will wait for corresponding idle indication before starting the power down flow.
16	CPU0Idle0Mask	RW 0x0	A value of 1 indicates that the PMU will wait for corresponding idle indication before starting the power down flow.
15	CPU0PwrDwnStts	RO 0x0	Indicates CPU0 is at power down 0 = PowerDown 1 = PowerUp
14:8	Reserved	RSVD 0x0	Reserved
7	CPU0Wake3Stts	RO 0x0	Reflects CPU0 Wake3 status
6	CPU0Wake2Stts	RO 0x0	Reflects CPU0 Wake2 status
5	CPU0Wake1Stts	RO 0x0	ReflectsCPU0 Wake1 status
4	CPU0Wake0Stts	RO 0x0	Reflects CPU0 Wake0 status
3	Reserved	RSVD 0x0	Reserved
2	CPU0Idle2Stts	RO 0x0	Reflects CPU0 Idle2 status
1	CPU0Idle1Stts	RO 0x0	Reflects CPU0 Idle1 status
0	CPU0Idle0Stts	RO 0x0	Reflects CPU0 Idle0 status

Table 272: CPU0 Idle Timer Value Register
Offset: 0x00022110

Bit	Field	Type/InitVal	Description
31:0	CPU0IdleTimerVal	RW 0x400	This value specifies the time period in which the idle indication will be periodically monitored. At the end of each time period, the counted idle indication will be compared to the configurable thresholds.

Table 273: CPU0 Idle High Threshold Register
Offset: 0x00022114

Bit	Field	Type/InitVal	Description
31:0	CPU0IdleHiThrs	RW 0x300	This value specifies the Idle high threshold. If the number of Idle indications, counted during an “Idle Timer” period, is greater then this threshold, a IdleHighStatus interrupt will be set.

Copyright © 2015 Marvell

Table 274: CPU0 Idle Low Threshold Register
Offset: 0x00022118

Bit	Field	Type/InitVal	Description
31:0	CPU0IdleLoThrs	RW 0x100	This value specifies the Idle low threshold. If the number of Idle indications, counted during an “Idle Timer” period, is less then this threshold, a IdleLowStatus interrupt will be set.

Table 275: CPU0 Idle Result Register
Offset: 0x0002211C

Bit	Field	Type/InitVal	Description
31:0	CPU0IdleResult	RW 0x0	This value specifies the number of Idle indication which were counted during the “Idle Timer” period. This field is updated after each “Idle Timer” period is completed. If the number of Idle indication, counted during an “Idle Timer” period exceeds the thresholds, the result value is locked, to enable SW to read it. The SW needs to write (clear) this register, to unlock it. Notes: - Write cannot be performed on this register unless the CPUn statistics counter is disabled (register 0x22100 + n*0x100, bit 0). To clear and unlock this register, disable the statistics counter and write 0. - The values do not get locked (stay 0) unless the idle counter is less than the low threshold (register 0x22118 + n*0x100) or greater than the high threshold (register 0x22114 + n*0x100).

Table 276: CPU0 PM Event Status and Mask Register
Offset: 0x00022120

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25	CPU0IdleLowThrshIntMask	RW 0x0	Mask control of the CPU0IdleLowThrshInt event, to generate a unit’s PM interrupt.
24	CPU0IdleHighThrshIntMask	RW 0x0	Mask control of the CPU0IdleHighThrshInt event, to generate a unit’s PM interrupt.
23:18	Reserved	RSVD 0x0	Reserved
17	CPU0DfsDoneMask	RW 0x0	Mask control of the CPU0DfsDone event, to generate a unit’s PM interrupt.
16	CPU0PoweredUpMask	RW 0x0	Mask control of the CPU0PoweredUp event, to generate a unit’s PM interrupt.
15:10	Reserved	RSVD 0x0	Reserved
9	CPU0IdleLowThrshInt	RW1C 0x0	CPU0 Idle statistics counter low threshold interrupt.

Copyright © 2015 Marvell

Table 276: CPU0 PM Event Status and Mask Register (Continued)
Offset: 0x00022120

Bit	Field	Type/InitVal	Description
8	CPU0IdleHighThrshInt	RW1C 0x0	CPU0 Idle statistics counter high threshold interrupt.
7:2	Reserved	RSVD 0x0	Reserved
1	CPU0DfsDone	RW1C 0x0	Set by the PMU to indicate that the DFS flow has completed.
0	CPU0PoweredUp	RW1C 0x0	Set by the PMU to indicate that the power up flow has completed.

Table 277: CPU0 Boot Address Redirect Register
Offset: 0x00022124

Bit	Field	Type/InitVal	Description
31:0	CPU0BootAddrRedirect	RW 0xffff0000	CPU0 Boot Address Redirect. Software can prepare in this register, the address to which CPU0 is required to boot from. The BootRom firmware should use this register content to redirect the boot address.

Table 278: CPU0 Resume Control Register
Offset: 0x00022128

Bit	Field	Type/InitVal	Description
31:0	ResumeControlInfo	RW 0x0	Resume Control Information, used by software to determine if and what actions needs to be performed before continue to execute from the Boot Address Redirect, when resuming from power down.

A.3.2 CPU1 Power Management

Table 279: CPU1 Statistic Counter Global Control Register
Offset: 0x00022200

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	CPU1StatCountEn	RW 0x0	CPU1 statistics counter enable control. 0 = Disable: Statistic counter disabled. 1 = Enable: Statistic counter enabled.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 280: CPU1 PM Control and Configuration Register
Offset: 0x00022204

Bit	Field	Type/InitVal	Description
31:19	Reserved	RSVD 0x0	Reserved
18	CPU1DfsReq	RW 0x0	CPU1 DFS request. Cleared by HW.
17	Reserved	RSVD 0x0	Reserved
16	CPU1PowerDownReq	RW 0x0	CPU1 power down request. Cleared by HW.
15:8	Reserved	RSVD 0x0	Reserved
7:4	CPU1FrequencyID	RW 0x1	Indicates CPU1 Frequency ID.
3:0	CPU1PowerDomainID	RW 0x2	Indicates CPU1 Power-Domain ID.

Table 281: CPU1 PM Status and Mask Register
Offset: 0x0002220C

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25	CPU1Intr1Mask	RW 0x0	Controls the masking control of the unit's wake1 indication, as an interrupt. A value of 1 means that this interrupt signals will be masked from the unit. 0 = NoMask: Do not mask the unit's interrupt. 1 = Mask: Mask the unit's interrupt.
24	CPU1Intr0Mask	RW 0x0	Controls the masking control of the unit's wake0 indication, as an interrupt. A value of 1 means that this interrupt signals will be masked from the unit. 0 = NoMask: Do not mask the unit's interrupt. 1 = Mask: Mask the unit's interrupt.
23	CPU1Wake3Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
22	CPU1Wake2Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
21	CPU1Wake1Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
20	CPU1Wake0Mask	RW 0x0	A value of 1 indicates that the corresponding wake event will trigger the power up flow.
19	Reserved	RSVD 0x0	Reserved
18	CPU1Idle2Mask	RW 0x0	A value of 1 indicates that the PMU will wait for corresponding idle indication before starting the power down flow.

Copyright © 2015 Marvell

Table 281: CPU1 PM Status and Mask Register (Continued)
Offset: 0x0002220C

Bit	Field	Type/InitVal	Description
17	CPU1Idle1Mask	RW 0x0	A value of 1 indicates that the PMU will wait for corresponding idle indication before starting the power down flow.
16	CPU1Idle0Mask	RW 0x0	A value of 1 indicates that the PMU will wait for corresponding idle indication before starting the power down flow.
15	CPU1PwrDwnStts	RO 0x0	Indicates CPU1 is at power down 0 = PowerDown 1 = PowerUp
14:8	Reserved	RSVD 0x0	Reserved
7	CPU1Wake3Stts	RO 0x0	Reflects CPU1 Wake3 status
6	CPU1Wake2Stts	RO 0x0	Reflects CPU1 Wake2 status
5	CPU1Wake1Stts	RO 0x0	ReflectsCPU1 Wake1 status
4	CPU1Wake0Stts	RO 0x0	Reflects CPU1 Wake0 status
3	Reserved	RSVD 0x0	Reserved
2	CPU1Idle2Stts	RO 0x0	Reflects CPU1 Idle2 status
1	CPU1Idle1Stts	RO 0x0	Reflects CPU1 Idle1 status
0	CPU1Idle0Stts	RO 0x0	Reflects CPU1 Idle0 status

Table 282: CPU1 Idle Timer Value Register
Offset: 0x00022210

Bit	Field	Type/InitVal	Description
31:0	CPU1IdleTimerVal	RW 0x400	This value specifies the time period in which the idle indication will be periodically monitored. At the end of each time period, the counted idle indication will be compared to the configurable thresholds.

Table 283: CPU1 Idle High Threshold Register
Offset: 0x00022214

Bit	Field	Type/InitVal	Description
31:0	CPU1IdleHiThrs	RW 0x300	This value specifies the Idle high threshold. If the number of Idle indications, counted during an “Idle Timer” period, is greater then this threshold, a IdleHighStatus interrupt will be set.

Copyright © 2015 Marvell

Table 284: CPU1 Idle Low Threshold Register
Offset: 0x00022218

Bit	Field	Type/InitVal	Description
31:0	CPU1IdleLoThrs	RW 0x100	This value specifies the Idle low threshold. If the number of Idle indications, counted during an “Idle Timer” period, is less then this threshold, a IdleLowStatus interrupt will be set.

Table 285: CPU1 Idle Result Register
Offset: 0x0002221C

Bit	Field	Type/InitVal	Description
31:0	CPU1IdleResult	RW 0x0	This value specifies the number of Idle indication which were counted during the “Idle Timer” period. This field is updated after each “Idle Timer” period is completed. If the number of Idle indication, counted during an “Idle Timer” period exceeds the thresholds, the result value is locked, to enable SW to read it. The SW needs to write (clear) this register, to unlock it. Notes: - Write cannot be performed on this register unless the CPUn statistics counter is disabled (register 0x22100 + n*0x100, bit 0). To clear and unlock this register, disable the statistics counter and write 0. - The values do not get locked (stay 0) unless the idle counter is less than the low threshold (register 0x22118 + n*0x100) or greater than the high threshold (register 0x22114 + n*0x100).

Table 286: CPU1 PM Event Status and Mask Register
Offset: 0x00022220

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25	CPU1IdleLowThrshIntMask	RW 0x0	Mask control of the CPU1IdleLowThrshInt event, to generate a unit’s PM interrupt.
24	CPU1IdleHighThrshIntMask	RW 0x0	Mask control of the CPU1IdleHighThrshInt event, to generate a unit’s PM interrupt.
23:18	Reserved	RSVD 0x0	Reserved
17	CPU1DfsDoneMask	RW 0x0	Mask control of the CPU1DfsDone event, to generate a unit’s PM interrupt.
16	CPU1PoweredUpMask	RW 0x0	Mask control of the CPU1PoweredUp event, to generate a unit’s PM interrupt.
15:10	Reserved	RSVD 0x0	Reserved
9	CPU1IdleLowThrshInt	RW1C 0x0	CPU1 Idle statistics counter low threshold interrupt.

Copyright © 2015 Marvell

Table 286: CPU1 PM Event Status and Mask Register (Continued)
Offset: 0x00022220

Bit	Field	Type/InitVal	Description
8	CPU1IdleHighThrshInt	RW1C 0x0	CPU1 Idle statistics counter high threshold interrupt.
7:2	Reserved	RSVD 0x0	Reserved
1	CPU1DfsDone	RW1C 0x0	Set by the PMU to indicate that the DFS flow has completed.
0	CPU1PoweredUp	RW1C 0x0	Set by the PMU to indicate that the power up flow has completed.

Table 287: CPU1 Boot Address Redirect Register
Offset: 0x00022224

Bit	Field	Type/InitVal	Description
31:0	CPU1BootAddrRedirect	RW 0xffff0000	CPU1 Boot Address Redirect. Software can prepare in this register, the address to which CPU1 is required to boot from. The BootRom firmware should use this register content to redirect the boot address.

Table 288: CPU1 Resume Control Register
Offset: 0x00022228

Bit	Field	Type/InitVal	Description
31:0	ResumeControlInfo	RW 0x0	Resume Control Information, used by software to determine if and what actions needs to be performed before continue to execute from the Boot Address Redirect, when resuming from power down.

Copyright © 2015 Marvell

A.4 NAND Flash Registers

The following table provides a summarized list of all registers that belong to the NAND Flash, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 289: Summary Map Table for the NAND Flash Registers

Register Name	Offset	Table and Page
NAND Flash		
Data Flash Control Register (NDCR)	0x000D0000	Table 290, p. 518
NAND Interface Timing Parameter 0 Register (NDTR0CS0)	0x000D0004	Table 291, p. 524
NAND Interface Timing Parameter 1 Register (NDTR1CS0)	0x000D000C	Table 292, p. 527
NAND Controller Status Register (NDSR)	0x000D0014	Table 293, p. 529
NAND Controller Page Count Register (NDPCR)	0x000D0018	Table 294, p. 534
NAND Controller Bad Block Registers (NDBBRx)	0x000D001C	Table 295, p. 535
NAND ECC Control (NDECCCTRL) Register	0x000D0028	Table 296, p. 535
NAND Busy Length Count (NDBZCNT) Register	0x000D002C	Table 297, p. 536
NAND Controller Data Buffer (NDDB) Register	0x000D0040	Table 298, p. 537
NAND Controller Command Buffer 0 (NDCB0) Register	0x000D0048	Table 299, p. 538
NAND Controller Command Buffer 1 (NDCB1) Register	0x000D004C	Table 300, p. 541
NAND Controller Command Buffer 2 (NDCB2) Register	0x000D0050	Table 301, p. 541
NAND Controller Command Buffer 3 (NDCB3) Register	0x000D0054	Table 302, p. 543

Copyright © 2015 Marvell

A.4.1 NAND Flash

Table 290: Data Flash Control Register (NDCR)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
Program the Timing Parameter registers (NDTR0CSx and NDTR1CSx) and Control registers before writing to this register. This register must be programmed before the partition registers are written to because writing any protection regions into the protection registers overrides updates to the PG_PER_BLK and RA_START fields. The reason for this blocking is to prevent an incorrect update of the control register that can “confuse” the partition address matching functions. This is a read/write register. Ignore reads from reserved bits. Write 0 to reserved bits.			
31	SPARE_EN	RW 0x0	Spare Area Enable Set the spare-area enable (SPARE_EN) bit to use the spare area of the NAND flash for data storage. If the SPARE_EN bit is set and ECC is not enabled (ECC_EN cleared), it is possible to write to and read from the entire data area and spare area. If SPARE_EN is clear, the spare area is not available to users. If SPARE_EN and ECC_EN are set, the remaining spare bytes after ECC is written are available to software. NDTR0CSx and NDTR1CSx registers specify the data bytes available for program/read, to the user per page for different SPARE_EN and ECC_EN settings. The ECCCTRL register defines the number of bytes used by ECC in ECCCTRL.ECC_SPARE. For Hamming ECC, the compatible definition is used. For extended ECC additional bytes are required to achieve the needed error recovery. 0 = Disable: Write/Read to available spare area disabled. 1 = Enable: Write/Read to available spare area enabled.
30	ECC_EN	RW 0x0	ECC Enable When the ECC enable (ECC_EN) bit is set, it enables the computation of ECC and error detection and correction. When ECC_EN bit is clear, ECC computation and error checks are not performed. The ECCCTRL register defines the type of ECC to be performed. The compatible algorithm is to use a single bit error correct, double bit detect Hamming code where a syndrome is generated for each 512 bytes. For extended correction a BCH algorithm with 16-bit per 2K byte page, correcting code is used. 0 = Disable: ECC is disabled for write and read data. 1 = Enable: ECC is enabled for write and read data.
29	DMA_EN	RW 0x0	DMA Request Enable When this bit is clear, no DMA requests are asserted. **NOTE:** When PDMA is set to OFF, this bit must be cleared to 0x0. 0 = Disable: Data and Command DMA are disabled. 1 = Enable: Data and Command DMA are enabled.
28	ND_RUN	RW 0x0	NAND Controller Run Mode When the NAND-controller run mode (ND_RUN) bit is set, the NAND controller starts to execute the command in the command buffer. When PDMA is set to ON, if the command buffer is empty when the ND_RUN bit is set, the NAND controller asserts the command-DMA request if the DMA request is enabled. The ND_RUN bit is cleared when the NAND flash controller finishes the execution of a command with the next command (NC) bit clear, indicating the end of the command sequence. For a bad-block detect, ND_RUN is cleared after the command buffer is emptied. Clearing the ND_RUN bit during a command execution results in an immediate termination of transactions to the flash device and clears the data and command buffers. 0 = Disable: NAND controller is not in run mode. 1 = Enable: NAND controller is in run mode.

Document Classification: Proprietary Information

Table 290: Data Flash Control Register (NDCR) (Continued)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
27	DWIDTH_C	RW 0x0	Data Bus Width of the NAND Flash Controller Flash-controller data width (DWIDTH_C) defines the data-bus width used by the NAND flash controller to communicate to the flash memory device(s). When DWIDTH_C = 1 and DWIDTH_M = 0, two flash devices are connected to the same device. 0 = 8-bits: Data Bus width is 8 bits. 1 = 16-bits: When 16b_NFC is set to ON, Data Bus width is 16 bits. Otherwise, this is reserved.
26	DWIDTH_M	RW 0x0	Data Bus Width of the NAND Flash Memory Flash-memory data width (DWIDTH_M) defines the data bus width of the NAND flash-memory devices used in the memory system. DWIDTH_M and DWIDTH_C determine the way NAND flash devices are interfaced. DWIDTH_C = 0 and DWIDTH_M = 1 is a non-valid setting. If used, the flash controller behavior defaults to DWIDTH_C = 0 and DWIDTH_M = 0. When DWIDTH_C = 1 and DWIDTH_M = 0, two flash devices are connected to the same device. 0 = 8-bits: Data Bus width is 8 bits. 1 = 16-bits: When 16b_NFC is set to ON, Data Bus width is 16 bits. Otherwise, this is reserved.
25:24	PAGE_SZ	RW 0x0	Page Size of the Flash device The page-size (PAGE_SZ) field defines the page size of the flash memory. All memory devices used in the flash-memory system must have the same bus width, page size, command set, and timing parameters.
23	SEQ_DIS	RW 0x0	Sequential page read disable This bit should be set if sequential page read is not wanted. When SEQ_DIS bit is clear, all multi-page reads are completed sequentially. The device is addressed only when the first page is read, and subsequent page reads are performed by toggling NF_REn when the device is ready. Consult the data sheet of the flash part under “Sequential Row Read” or similar for details. **NOTE:** Sequential read is supported only when the entire page (including the spare area) is read by the NAND flash controller. The cases where sequential read is supported are: {PAGE_SZ = 00, SPARE_EN = 1, ECC_EN = 0}, {PAGE_SZ = 01, SPARE_EN = 1, ECC_EN = 0} and {PAGE_SZ = 01, SPARE_EN = 1, ECC_EN = 1}. For all other settings of PAGE_SZ, SPARE_EN and ECC_EN sequential read are not supported, the NCSX bit must be set, and multi-page reads are performed after addressing each page individually. If SEQ_DIS is clear, in between multiple read operations will wait for NDTR1CS0.tR_NFC before transferring more data. If NDTR1CS0.WAIT_MODE is set, the value in NDTR1CS0.tR_NFC can be “small” and the NFC will wait for a minimum time before transferring more data. If WAIT_MODE is not set, the value in NDTR1CS0.tR_NFC will need to be set to the tR parameter from the NAND flash device’s data sheet. In this case the data throughput may not be optimal. 0 = No Command Cycle: Sequential page reads (without commands) are supported. 1 = Command Cycle Required: All sequential page reads require command cycles.
22	Reserved	RSVD 0x0	Reserved

Table 290: Data Flash Control Register (NDCR) (Continued)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
21	FORCE_CSX	RW 0x0	Force chip select false on busy If the FORCE_CSX bit is set, then the NF_CSn for the addressed flash must be de-asserted during the time where Busy is expected to be asserted. In addition, during all operations DFI (Data Flash Interface) arbitration must be allowed during the command and/or data transfer phases. Under normal operation, NF_CSn shall stay asserted throughout the command and also throughout data transfer phases, but if the DFI is being arbitrated away from the NFC, the command sequence or data sequence shall be interrupted at the beginning of the current cycle by first de-asserting NF_CSn and then relinquishing the DFI to the other slave. During an address phase where NF_ALE is asserted, NF_ALE is de-asserted along with NF_CSn. When the NFC reacquires the DFI, NF_ALE and NF_CSn will assert and address cycles will continue with NF_WEn strobes. When NOR/NAND hardware arbiter enabled, this bit must be set. 0 = Assert NF_CSn: Assert chip select during entire command, except during command overlap. 1 = De-assert NF_CSn: De-assert chip select during busy phases.
20	CLR_PG_CNT	RW 0x0	Clear Page Count The clear-page-count (CLR_PG_CNT) bit, when set, clears the Page Count register. After the NAND Interface Page Count register is cleared, the CLR_PG_CNT bit is reset. When a program failure occurs, CLR_PG_CNT can be used to clear the page count values after software reads the number of pages programmed correctly. 0 = Not Cleared: Page count not cleared 1 = Clear: Clear page count
19	STOP_ON_UNCOR	RW 0x0	Stop On Uncorrectable Error If an uncorrectable error is received on a read and this bit is set, NDCB0.NC is reset and when PDMA is set to ON, further command DMA processing is halted. 0 = Continue: Do not stop on uncorrectable error. 1 = Stop: Stop on uncorrectable error.
18:16	RD_ID_CNT	RW 0x0	Read ID Byte Count The read-ID byte-count (RD_ID_CNT) bit specifies the number of data bytes to read from the flash device when a read-ID command is issued. Valid values for RD_ID_CNT are 1–7. For a read ID command, the page count should be set to 1 (NDCB2[Page_Count] should be zero). Values from 1 to 7. Specifies the number of bytes of read ID data to be read from the flash device
15	RA_START	RW 0x0	Row Address Start Position The row-address start-position (RA_START) bit specifies the address cycle where row address starts in the addressing sequence. When RA_START is clear, NDCB1[ADDR2] (address sent out in the second addressing cycle) contains the first row address. When RA_START is set, NDCB1[ADDR3] (address sent out in the third addressing cycle) contains the first row address. 0 = 2nd Address Cycle: Row address supplied to flash from second address cycle onwards 1 = 3rd Address Cycle: Row address supplied to flash from third address cycle onwards

Copyright © 2015 Marvell

Table 290: Data Flash Control Register (NDCR) (Continued)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
14:13	PG_PER_BLK	RW 0x0	Pages Per Block The pages-per-block (PG_PER_BLK) bit specifies the number of pages in one block of the flash device. The PG_PER_BLK bit, when cleared, indicates 32 pages/block, and when set, indicates 64 pages/block. PG_PER_BLK along with RA_START is used by the NAND flash controller to increment the row address in multi-page commands. 00 = Flash device has 32 pages per block 10 = Flash device has 64 pages per block 01 = Flash device has 128 pages per block 11 = Flash device has 256 pages per block **NOTE:** The order of bits seems backwards due to compatibility. Previously bit 13 was reserved.
12	Reserved	RSVD 0x1	Reserved
11	RDYM	RW 0x1	Flash Device Ready Interrupt Mask The flash-device ready-mask (RDYM) bit is used to mask an interrupt request that is asserted when NF_RBn (Ready/Busy_) makes a transition from low to high, as indicated by NDSR.RDY. When RDYM is cleared, the interrupt is enabled, and whenever NDSR[RDY] is set, an interrupt request is made to the interrupt controller. When RDYM is cleared, the interrupt is masked and the state of the RDY status bit does not generate an interrupt. If more than one NF_RBn signal is used, this mask bit enables/disables interrupts for both. 0 = Enable: Flash device ready interrupt is enabled. 1 = Disable: Flash device ready interrupt is disabled.
10	CS0_PAGEDM	RW 0x1	NF_CSn[0,2] Page Done Interrupt Mask The CS0 page-done mask (CS0_PAGEDM) bit is used to mask or enable an interrupt request that is asserted when a page transaction (read or write) to a flash device interfaced using NF_CSn[0] is completed, as indicated by NDSR[CS0_PAGED]. When CS0_PAGEDM is cleared, the interrupt is enabled; whenever NDSR[CS0_PAGED] is set, an interrupt request is made to the interrupt controller. When CS0_PAGEDM is set, the interrupt is masked and the state of the PAGE_DN status bit does not generate an interrupt. **NOTE:** All reference to NF_CSn[2] is only applicable when 16b_NFC is set to ON. 0 = Enable: NF_CSn[0,2] page done interrupt is enabled. 1 = Disable: NF_CSn[0,2] page done interrupt is disabled.
9	CS1_PAGEDM	RW 0x1	NF_CSn[1,3] Page Done Interrupt Mask When 16b_NFC is set to ON, CS1 Page Done Mask (CS1_PAGEDM) is used to mask or enable an interrupt request that is asserted when a page transaction (read or write) to flash device interfaced using NF_CSn[1] is completed as indicated by NDSR[CS1_PAGED]. When CS1_PAGEDM is cleared, the interrupt is enabled, and whenever NDSR[CS1_PAGED] is set, an interrupt request is made to the interrupt controller. When CS1_PAGEDM is set, the interrupt is masked and the state of the PAGE_DN status bit does not generate an interrupt. **NOTE:** When 16b_NFC is set to OFF, this bit must be set to 1. 0 = Enable: NF_CSn[1,3] page done interrupt is enabled. 1 = Disable: NF_CSn[1,3] page done interrupt is disabled.

Copyright © 2015 Marvell

Table 290: Data Flash Control Register (NDCR) (Continued)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
8	CS0_CMDDM	RW 0x1	NF_CSn[0,2] Command Done Interrupt Mask The CS0 command-done mask (CS0_CMDDM) bit is used to mask or enable an interrupt request that is asserted when the command for the flash device interfaced using NF_CSn[0] has been executed, as indicated by NDSR[CS0_CMDD]. When CS0_CMDDM is cleared, the interrupt is enabled; whenever NDSR[CS0_CMDD] is set, an interrupt request is made to the interrupt controller. When CS0_CMDDM is set, the interrupt is masked and the state of the CS0_CMDD status bit does not generate an interrupt. **NOTE:** All reference to NF_CSn[2] is only applicable when 16b_NFC is set to ON. 0 = Enable: NF_CSn[0,2] command done interrupt is enabled. 1 = Disable: NF_CSn[0,2] command done interrupt is disabled.
7	CS1_CMDDM	RW 0x1	NF_CSn[1,3] Command Done Interrupt Mask When 16b_NFC is set to ON, the CS1 command-done mask (CS1_CMDDM) bit is used to mask or enable an interrupt request that is asserted when the command for the flash device interfaced using ND_nCS1 has been executed, as indicated by NDSR[CS1_CMDD]. When CS1_CMDDM is cleared, the interrupt is enabled, and whenever NDSR[CS1_CMDD] is set, an interrupt request is made to the interrupt controller. When CS1_CMDDM is set, the interrupt is masked and the state of the CS1_CMDD status bit does not generate an interrupt. **NOTE:** When 16b_NFC is set to OFF, this bit must be set to 1. 0 = Enable: NF_CSn[1,3] command done interrupt is enabled. 1 = Disable: NF_CSn[1,3] command done interrupt is disabled.
6	CS0_BBDM	RW 0x1	NF_CSn[0,2] Bad Block Detect Interrupt Mask The CS0 bad-block-detect mask (CS0_BBDM) bit is used to mask or enable an interrupt request that is asserted when the status check at the end of a page write or block erase to a device interfaced using ND_CSn[0,2] returns a bad-block error, as indicated by NDSR[CS0_BBD]. When CS0_BBDM = 0, the interrupt is enabled; whenever NDSR[CS0_BBD] is set, an interrupt request is made to the interrupt controller. When CS0_BBDM=1, the interrupt is masked and the state of the CS0_BBD status bit does not generate an interrupt. **NOTE:** All reference to NF_CSn[2] is only applicable when 16b_NFC is set to ON. 0 = Enable: NF_CSn[0,2] bad block detect interrupt is enabled. 1 = Disable: NF_CSn[0,2] bad block detect interrupt is disabled.

Copyright © 2015 Marvell

Table 290: Data Flash Control Register (NDCR) (Continued)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
5	CS1_BBDM	RW 0x1	NF_CSn[1,3] Bad Block Detect Interrupt Mask When 16b_NFC is set to ON, CS1 Bad Block Detect Mask (CS1_BBDM) is used to mask or enable an interrupt request that is asserted when the status check at the end of a page write or block erase to a device interfaced using NF_CSn[1,3] returns a bad block error, as indicated by NDSR[CS1_BBD]. When CS1_BBDM is cleared, the interrupt is enabled, and whenever NDSR[CS1_BBD] is set, an interrupt request is made to the interrupt controller. When CS1_BBDM is set, the interrupt is masked and the state of the CS1_BBD status bit does not generate an interrupt. **NOTE:** For the reliable operation of flash media, always enable CS0_BBD and CS1_BBD interrupts (CS0_BBDM and CS1_BBDM cleared), and have software perform block replacement based on these interrupts. When 16b_NFC is set to OFF, this bit must be set to 1. 0 = Enable: NF_CSn[1,3] bad block detect interrupt is enabled. 1 = Disable: NF_CSn[1,3] bad block detect interrupt is disabled.
4	UNCERRM	RW 0x1	ECC Failure Error Interrupt Mask The ECC Failure error mask (UNCERRM) bit is used to mask or enable an interrupt request that is asserted when an uncorrectable Hamming double-bit error is detected in any of the data streams as indicated by NDSR[UNCERR]. When in extended ECC mode, this bit indicates a failure with the BCH ECC. When UNCERRM is cleared, the interrupt is enabled; whenever NDSR[UNCERR] is set, an interrupt request is made to the interrupt controller. When UNCERRM is set, the interrupt is masked and the state of the UNCERR status bit does not generate an interrupt. No data correction is possible for either kind of failure. 0 = Enable: ECC Failure error interrupt is enabled. 1 = Disable: ECC Failure error interrupt is disabled.
3	CORERRM	RW 0x1	Correctable Error Interrupt Mask The correctable error mask (CORERRM) bit is used to mask or enable an interrupt request that is asserted when either a single-bit error is detected in any of the data streams when in Hamming ECC mode) or when the number of error exceeds the threshold defined in the ECCCTRL register when in BCH mode. This condition is indicated by NDSR[CORERR]. When CORERRM is cleared, the interrupt is enabled; whenever NDSR[CORERR] is set, an interrupt request is made to the interrupt controller. When CORERRM is set, the interrupt is masked and the state of the UNCERR status bit does not generate an interrupt. This interrupt is an indication only and can be used to determine if read disturbs require a block to be rewritten. 0 = Enable: Single-bit error interrupt is enabled. 1 = Disable: Single-bit error interrupt is disabled.
2	WRDREQM	RW 0x1	Write Data Request Interrupt Mask The write-data-request mask (WRDREQM) bit is used to mask or enable an interrupt request that is asserted when a write command is loaded into command buffer and data buffer is empty, as indicated by NDSR[WRDREQ]. When WRDREQM is cleared, the interrupt is enabled; whenever NDSR[WRDREQ] is set, an interrupt request is made to the interrupt controller. When WRDREQM is set, the interrupt is masked and the state of the WRDREQ status bit does not generate an interrupt. 0 = Enable: Write data request interrupt is enabled. 1 = Disable: Write data request interrupt is disabled.

Copyright © 2015 Marvell

Table 290: Data Flash Control Register (NDCR) (Continued)
Offset: 0x000D0000

Bit	Field	Type/InitVal	Description
1	RDDREQM	RW 0x1	Read Data Request Interrupt Mask Read data request mask (RDDREQM) is used to mask or enable an interrupt request that is asserted when the data buffer has been loaded with a page of read data, as indicated by NDSR[RDDREQ]. When RDDREQM is cleared, the interrupt is enabled, and whenever NDSR[RDDREQ] is set, an interrupt request is made to the interrupt controller. When RDDREQM is set, the interrupt is masked and the state of the RDDREQ status bit does not generate an interrupt. 0 = Enable: Read data request interrupt is enabled. 1 = Disable: Read data request interrupt is disabled.
0	WRCMDREQM	RW 0x1	Write Command Request Interrupt Mask Write command request mask (WRCMDREQM) is used to mask or enable an interrupt request that is asserted when a write to the command buffer is requested, as indicated by NDSR[WRCMDREQ]. When WRCMDREQM is cleared, the interrupt is enabled, and whenever NDSR[WRCMDREQ] is set, an interrupt request is made to the interrupt controller. When WRCMDREQM is set, the interrupt is masked and the state of the RDDREQ status bit does not generate an interrupt. 0 = Enable: Write command request interrupt is enabled. 1 = Disable: Write command request interrupt is disabled.

Table 291: NAND Interface Timing Parameter 0 Register (NDTR0CS0)
Offset: 0x000D0004

Bit	Field	Type/InitVal	Description
Program this register to configure the setup and hold times of outputs driven by the NAND flash controller (NF_ALE, NF_CLE, NF_CSn) and the pulse widths and cycle times of NF_REn and NF_WEn. Accesses to NAND flash devices interfaced using NF_CSn are controlled by the settings in NDTR0CS0. Programmable timing parameters enable the interface to a wide variety of NAND flash devices. All timing parameters in NDTR0CS0 are set in terms of NAND controller clock periods. The NAND controller clock runs at the ND_CLK frequency. Refer to the Hardware Specifications for frequency limitations of the NAND flash controller timing parameters. The NDTR0CS0 and NDTR1CS0 registers define the timings as integer multiples of the NFC clock. This is a read/write register. Ignore reads from reserved bits. Write 0 to reserved bits.			
31:27	tADL_NFC	RW 0x1	Address to Write Data Delay This parameter controls the time between the rising edge of the NF_WEn strobe for the last address cycle until the first rising edge of the NF_WEn strobe for a write operation. Value is from 0 to 31 and specifies in terms of ND_CLK periods. The fields must be set to the difference between the NAND's device tADL requirement and the Max(tWH_NFC,tCH_NFC) * t(ND_CLK) ns. Requirement (tADL refer to the parameters of the NAND flash data sheet): tADL_NFC * t(ND_CLK) > (tADL - Max(tWH_NFC,tCH_NFC) * t(ND_CLK))
26	SelCntr	RW 0x1	Select Read Counter This bit determines the source of the read strobe. SW must set this field to "Strobe". 0 = Reserved 1 = Strobe: Strobe is Rd_Cnt_Del+1 ND_CLK clocks from the rising edge of NF_REn.

Copyright © 2015 Marvell

Table 291: NAND Interface Timing Parameter 0 Register (NDTR0CS0) (Continued)
Offset: 0x000D0004

Bit	Field	Type/InitVal	Description
25:22	Rd_Cnt_Del	RW 0x8	Read Strobe Count Delay In read commands, this counter used to latch the read data into the NFC relatively to the NF_REn edge. The NF_REn falling or rising edge is the trigger to this counter (The edge can be selected by sel_nRE_edge field). This value is the number of ND_CLK clocks after the rising/falling edge of NF_REn to latch read data into the NFC from the DFI.
21:19	tCH_NFC	RW 0x4	Enable Signal Hold Time Enable signal hold time (tCH) defines the hold time (with respect to the rising edge of NF_WEn) of NF_CSn, NF_ALE and NF_CLE. Value from 0 to 7. Specifies the hold time for NF_CSn, NF_ALE, and NF_CLE outputs in terms of NAND controller clock periods. Requirement (tCH, tCLH and tALH refer to the parameters of the NAND flash data sheet): (tCH_NFC + 1) * t(ND_CLK) > Max(tCH, tCLH, tALH)
18:16	tCS_NFC	RW 0x1	Enable Signal Setup Time Enable signal setup time (tCS) defines the setup time (with respect to the falling edge of NF_WEn) of NF_CSn, NF_ALE and NF_CLE. Value from 0 to 7. Specifies the setup time for NF_CSn, NF_ALE, and NF_CLE outputs in terms of NAND controller clock periods. Requirements (tCS, tCLS, tALS, tREA and tCEA refer to the parameters of the NAND flash data sheet): 1. (tCS_NFC + 1) * t(ND_CLK) >= tCS - (tWP_NFC + 1) * t(ND_CLK) 2. (tCS_NFC + 1) * t(ND_CLK) >= Max(tCLS,tALS) - (tWP_NFC + 1) * t(ND_CLK) 3. When FORCE_CSX enabled: (tCS_NFC +1) * t(ND_CLK) >= (tCEA - tREA)
15:14	Reserved	RSVD 0x0	Reserved
13:11	tWH_NFC	RW 0x1	NF_WEn high duration The period for which NF_WEn remains high, while address or data is being input to the NAND flash device in multiple NF_WEn cycles, is specified by NF_WEn high duration (tWH). The valid values for tWH_NFC are 0 through 7. Actual high duration of NF_WEn is (tWH_NFC +1) * t(ND_CLK). Value from 0 to 7. Specifies the NF_WEn high duration, in terms of ND_CLK periods. NF_WEn high duration = tWH_NFC+1 for a series of write pulses. For the last write pulse of a command, the duration is max(tWH_NFC,tCH_NFC)+1. Also note that this is the minimum duration that is guaranteed. The maximum duration for which the NF_WEn is high is greater than (tWH_NFC + 1) * t(ND_CLK) if the DFI bus is granted to the NOR during this time. Requirements (tWH and tWC refer to the parameters of the NAND flash data sheet): 1. (tWH_NFC + 1)*t(ND_CLK) >= tWH 2. (tWH_NFC + 1)*t(ND_CLK) >= tWC - (tWP_NFC + 1)*t(ND_CLK) 3. If NDCR.ECC_EN is true and NDECCCTRL.BCH_EN is true, tWH_NFC + tWP_NFC must always be greater or equal to 2.

Table 291: NAND Interface Timing Parameter 0 Register (NDTR0CS0) (Continued)
Offset: 0x000D0004

Bit	Field	Type/InitVal	Description
10:8	tWP_NFC	RW 0x1	NF_WEn pulse width The period for which NF_WEn is asserted low is specified in WE_Pulse width (tWP). The valid values for tWP_NFC are 0 through 7. Actual high duration of NF_WEn pulse would be (tWH_NFC +1) * t(ND_CLK). The NF_WEn cycle time is (tWH_NFC+tWP_NFC+2)*t(ND_CLK). Value from 0 to 7. Specifies the NF_WEn pulse width, in terms of NAND controller clock periods. NF_WEn pulse width = tWP_NFC + 1. Requirements (tCS, tCLS, tALS, tREA and tCEA refer to the parameters of the NAND flash data sheet): 1. (tWP_NFC + 1)*t(ND_CLK) >= tWP 2. (tWP_NFC + 1)*t(ND_CLK) >= tWC - (tWP_NFC + 1)*t(ND_CLK) 3. If NDCR.ECC_EN is true and NDECCCTRL.BCH_EN is true, tWH_NFC + tWP_NFC must always be greater or equal to 2.
7	sel_nRE_edge	RW 0x1	Select Read Strobe edge for Read data capture. This configuration select the relative edge that the Read Strobe Count Delay will start to count. 0 = Rising Edge: Select rising edge of NF_REn as reference to latch read data 1 = Falling Edge: Select falling edge of NF_REn as reference to latch read data.
6	etRP_NFC	RW 0x1	Extended tRP_NFC This bit is the MSB for the tRP_NFC value.
5:3	tRH_NFC	RW 0x1	NF_REn high duration The period for which NF_REn remains high while NF_REn is being toggled to access data during a read operation is specified by NF_REn high duration (tRH). The valid values for tRH_NFC are 0 through 7. Actual high duration of NF_REn is (tRH_NFC +1) * t(ND_CLK). Value from 0 to 7. Specifies the NF_REn high duration, in terms of NAND controller clock periods. NF_REn high duration = tRH_NFC+1. **NOTE:** This is the minimum duration that is guaranteed. The maximum duration for which the NF_REn is high is greater than ((tRH_NFC + 1)*t(ND_CLK)) if the DFI bus is granted to SMC during this time. Requirements (tRH and tRC refer to the parameters of the NAND flash data sheet): 1. (tRH_NFC + 1)*t(ND_CLK) >= tRH 2. (tRH_NFC + 1)*t(ND_CLK) >= tRC - ({etRP_NFC,tRP_NFC}+1)*t(ND_CLK) 3. If NDCR.ECC_EN is true and NDECCCTRL.BCH_EN is true, tRH_NFC + {etRP_NFC,tRP_NFC} must always be greater or equal to 2.

Copyright © 2015 Marvell

Table 291: NAND Interface Timing Parameter 0 Register (NDTR0CS0) (Continued)
Offset: 0x000D0004

Bit	Field	Type/InitVal	Description
2:0	tRP_NFC	RW 0x1	NF_REn pulse width These 3 bits in conjunction with bit 6 above (etRP_NFC) determine the period for which NF_REn is asserted low. The valid values for tRP_NFC are 0 through 16. Actual high duration of NF_REn pulse is {etRP_NFC,tRP_NFC}+1 NAND controller clock cycles. The NF_REn cycle time would be tRH_NFC+tRP_NFC+2 NAND controller clock cycles. Value from 0 to 16. Specifies the NF_REn pulse width, in terms of NAND controller clock periods. NF_REn pulse width = {etRP_NFC, tRP_NFC} +1. Requirements (tRP and tRC refer to the parameters of the NAND flash data sheet): 1. ({etRP_NFC,tRP_NFC} + 1)*t(ND_CLK) >= tRP 2. ({etRP_NFC,tRP_NFC} + 1)*t(ND_CLK) >= tRC - (tRH_NFC + 1) *t(ND_CLK) 3. If NDCR.ECC_EN is true and NDECCCTRL.BCH_EN is true, tRH_NFC + {etRP_NFC,tRP_NFC} must always be greater or equal to 2.

Table 292: NAND Interface Timing Parameter 1 Register (NDTR1CS0)
Offset: 0x000D000C

Bit	Field	Type/InitVal	Description
This register can be programmed to set the timing information for read, status read, and read-ID commands. Accesses to NAND flash devices interfaced using NF_CEn are controlled by the settings in NDTR1CS0. All timing parameters in NDTR1CS0 are set in terms of NAND controller-clock periods. The NAND controller clock runs at the frequency of the NFC clock (ND_MCLK). Refer to the Hardware Specifications for frequency limitation of the NAND flash controller timing parameters. When 16b_NFC is set to ON, although there is only one definition register, because it is possible to overlap operations on the chip selects, there are two actual timers, one for chip select 0, 2 and one for chip select 1, 3. This is a read/write register. Ignore reads from reserved bits. Write 0 to reserved bits.			
31:16	tR_NFC	RW 0x1	NF_WEn high to NF_REn Low for Read Value from 1 to 65535. Specifies additional delay between NF_WEn high and NF_REn low for read, in terms of NAND controller clock periods. This delay depends on the value programmed in NDTR0CS0.tCH_NFC in addition to tR_NFC programmed value. Requirements (tR and tWB refer to the parameters of the NAND flash data sheet). 1. (tR_NFC+2)*t(ND_CLK) >= tR - (tCH_NFC + 1)*t(ND_CLK) 2. If NDTR1CS0.WAIT_MODE is true then (tR_NFC >= tWB)

Table 292: NAND Interface Timing Parameter 1 Register (NDTR1CS0) (Continued)
Offset: 0x000D000C

Bit	Field	Type/InitVal	Description
15	Wait_Mode	RW 0x0	Determines how the information in this register is used to define the state machine transitions on read operations. For all reads, the timer always counts after the read command has been issued and this bit only determines the actual state machine proceed condition. **NOTE:** This bit may be changed from a 1 to a 0 while an operation is in progress. The reason for this is to provide a mechanism to reset in the event of RBn deadlock. If for some reason, RBn should stay de-asserted, resetting this bit after the time has expired will allow the state machine to proceed. This will release the controller and the NAND device should be issued a Reset command. In the event of RBn lockup, the result of the read operation is not defined. 0 = Specific: The tR_NFC time value specified in this register solely determines the duration. When this count has not expired, the state machine stalls and when the count reaches tR_NFC, regardless of the state of RBn, the operation will proceed. In other words RBn is ignored. 1 = Minimum: The tR_NFC time value specified in this register determines the minimum duration. When this count has not expired, the state machine stalls. When this count has expired, there may be an additional stall if RBnx is still indicating busy. The proceed condition is the AND of the time out from the defined timer and RBnx, for example, to proceed, both the time value must have expired and RBnx must be signaling “ready”.
14	PRESCALE	RW 0x0	Prescale tR_NFC Regardless of the setting of the PRESCALE bit, a tR_NFC value of zero produces a minimum of ND_CLK clock delay. A value for tR_NFC of 1 produces a two ND_CLK clock additional delay without PRESCALE and a 17 ND_CLK clock additional delay with PRESCALE. 0 = Whole NFC Clocks: The tR_NFC time value specified in this register is the number of NFC clocks to wait to begin to sample tR (in Wait_Mode) or to simply wait (when not in Wait_Mode). 1 = Fraction of NFC Clocks: The tR_NFC time value specified in this register is 1/16 of the number of NFC clocks i.e. the tR_NFC value is prescaled by 16.
13:10	Reserved	RSVD 0x0	Reserved
9:8	tRHW_NFC	RW 0x0	Read to Write Delay The tRHW parameter is the delay between the rising edge of a NF_REn strobe to the falling edge of a NF_WEn strobe. This situation occurs when a read command (of any type) is followed by another command. The NF_REn generates a read cycle that must be completely removed from the bus (i.e. tri-state) before the host can drive a command to write. The value of this field is multiplied by 16 to determine the number of clocks to guarantee between reads and the next command. 0 = None: 0 ND_CLK periods added 1 = 16: 16 ND_CLK periods added 2 = 32: 32 ND_CLK periods added 3 = 48: 48 ND_CLK periods added

Copyright © 2015 Marvell

Table 292: NAND Interface Timing Parameter 1 Register (NDTR1CS0) (Continued)
Offset: 0x000D000C

Bit	Field	Type/InitVal	Description
7:4	tWHR_NFC	RW 0x1	NF_WEn High to NF_REn Low for a Read Status For a status-read command, NF_WEn high to NF_REn Low Delay (tWHR) specifies the delay between de-assertion of NF_WEn and assertion of NF_REn. Value from 0 to 15. Specifies the delay between NF_CLE high and NF_REn low for status read, in terms of ND_CLK periods. Requirement (tWHR refer to the parameter of the NAND flash data sheet). The actual delay depends on tAR_NFC, tWH_NFC, tWHR_NFC, and tCH_NFC: (max(tWH_NFC, tCH_NFC) + max(tAR_NFC, max(0, tWHR_NFC-max(tWH_NFC,tCH_NFC))) + 2) * t(ND_CLK) >= tWHR
3:0	tAR_NFC	RW 0x1	NF_ALE Low to NF_REn Low Delay For a Read-ID command, NF_ALE Low to NF_REn Low delay (tAR) specifies the delay between de-assertion of NF_ALE and assertion of NF_REn. Requirement (tAR refer to the parameter of the NAND flash data sheet). The actual delay depends on tAR_NFC, tWH_NFC, tWHR_NFC, and tCH_NFC: (max(tWH_NFC, tCH_NFC) + max(tAR_NFC, max(0, tWHR_NFC-max(tWH_NFC,tCH_NFC))) + 2) * t(ND_CLK) >= tAR

Table 293: NAND Controller Status Register (NDSR)
Offset: 0x000D0014

Bit	Field	Type/InitVal	Description

This register contains bits that signal flash ready status, bad block detected, read-data-error conditions, data-read/write requests, completion of a page, and command. Unless masked, each of these hardware-detected events signals an interrupt request to the interrupt controller. Once set, each of the status bits is cleared by writing 0x1 to the corresponding bit position. Writing 0x0 has no effect on the status bits. Each of the NDSR bits signals an interrupt request as long as the bit is set and the corresponding interrupt is not masked. Once the bit is cleared, the interrupt is cleared.
NDSR status bits are set in both DMA and non-DMA modes of operation, and they can be polled and cleared by software in either mode.
This is a read/write register. Ignore reads from reserved bits. Write 0x0 to reserved bits.
NOTE: Care must be taken to only clear bits known to be set because it is possible that between the NDSR read and the subsequent write to clear the status, additional conditions will be set. If NDCR.ND_RUN is true, it is almost never correct to simply write 0x9FFF to the register (i.e. reset all bits). Doing so will most likely create the situation where the next expected event happens to assert just before the write to clear a previous event. This next event then gets cleared, and a software deadlock will occur.
The following examples show how the status register should be cleared.
NDSR = NDSR_WRCMDREQ; // reset the write command request status
NDSR = NDSR_WRCMDREQ | NDSR_CS0_CMDD; // reset command req and cmdd
NDSR = *NDSR; // reset all currently set status bits			

31:21	Reserved	RSVD 0x0	Reserved

Table 293: NAND Controller Status Register (NDSR) (Continued)
Offset: 0x000D0014

Bit	Field	Type/InitVal	Description
20:16	ERR_CNT	RO 0x0	Block Error Count If NDCR.ECC_EN is true, NDECCCTRL.BCH_EN is true an ECC check has been performed and (ERR_CNT > NDECCCTRL.ECC_THRESH), these bits indicate the number of errors encountered. If ECC_EN or BCH_EN are false, or (ERR_CNT <= ECC_THRESH) this field is 0x00. The error count is the number of non-zero syndromes. Writing 0x1f into this field will reset the error count. If the ERR_CNT field is 0x1f the errors are not correctable and UNCERR will also be set. **NOTE:** If it is desired to see the BCH error count in all cases, it will be necessary to set the ECC_THRESH to 0.
15	TRUSTVIO	RW 0x0	Trust Violation This bit is set when a trusted partition (or illegal operation) access violation occurs as defined by the NDCMDMAT{0,1,2} and NDPT{0..3}CS{0,1} registers.
14:13	Reserved	RSVD 0x0	Reserved
12	RDY1	RW 0x0	NAND Flash Ready CS1 The NAND flash-ready (RDY) status bit is set when NF_RBn[1] (NAND Flash Ready/Busy) has made a low-to-high transition, indicating the ready state entry of flash device(s). The NAND flash-ready condition can be programmed to cause an interrupt by clearing the flash-device-ready interrupt mask (RDYM). This status bit is updated continuously each time NF_RBn[1] input makes a low-to-high transition, regardless of the type of command sent to the flash device. When 16b_NFC is set to ON, in a ganged configuration (two 8-bit parts connected as a logical 16-bit part, where the RBn signals of each part are wire ANDed) the RDY reflects the worst case time from the two ganged parts. **NOTE:** Although the reset value of RDY1 is 0, this bit will be updated as soon as the NFC interface clock begins pulsing so that NDSR will most likely be set to 0x00001800 by the time it is captured. Both bits [12] and [11] are set together and should be reset together when there is not dual NF_RBn pins. 0 = No Transtition: The Ready/Busy_ input has not transitioned from low to high. 1 = Transition: The Ready/Busy_ input has transitioned from low to high.
11	RDY0	RW 0x0	NAND Flash Ready CS0 The NAND flash-ready (RDY) status bit is set when NF_RBn[0] (NAND Flash Ready/Busy) has made a low-to-high transition, indicating the ready state entry of flash device(s). The NAND flash-ready condition can be programmed to cause an interrupt by clearing the flash-device-ready interrupt mask (RDYM). This status bit is updated continuously each time NF_RBn[0] input makes a low-to-high transition, regardless of the type of command sent to the flash device. **NOTE:** Although the reset value of RDY0 is 0, this bit will be updated as soon as the NFC interface clock begins pulsing so that NDSR will most likely be set to 0x00001800 by the time it is captured. In a ganged configuration (two 8-bit parts connected as a logical 16-bit part, where the NF_RBn signals of each part are wire ANDed) the RDY reflects the worst case time from the two ganged parts. 0 = No Transtition: The Ready/Busy_ input has not transitioned from low to high. 1 = Transition: The Ready/Busy_ input has transitioned from low to high.

Copyright © 2015 Marvell

Table 293: NAND Controller Status Register (NDSR) (Continued)
Offset: 0x000D0014

Bit	Field	Type/InitVal	Description
10	CS0_PAGED	RW 0x0	NF_CSn[0,2] Page Done The CS0 page-done (CS0_PAGED) status bit is set when the read or write transaction to flash device(s) interfaced using NF_CSn[0,2] has completed a page of access including any user-defined spare-area access and ECC. CS0_PAGED is not set if an auto-status check after a program command results in a failure. CS0_PAGED bit is not set for read ID, read status, and erase commands. Page-done condition can be programmed to cause an interrupt by clearing the page- done interrupt mask (CS0_PAGEDM). 0 = No Boundary: The current read/write transaction on NF_CEn[0] has not reached a page boundary. 1 = Boundary: The current read/write transaction on NF_CEn[0] has reached a page boundary.
9	CS1_PAGED	RW 0x0	NF_CEn[1,3] Page Done The NF_CEn[1,3] page-done (CS1_PAGED) status bit is set when the read or write transaction to flash device(s) interfaced using NF_CEn[1,3] has completed a page of access including any user- defined spare-area access and ECC. CS1_PAGED is not set if an auto-status check after a program command results in a failure. The CS1_PAGED bit is not set for read ID, read status, and erase commands. Page-done condition can be programmed to cause an interrupt by clearing the page- done interrupt mask (CS1_PAGEDM). **NOTE:** Page-done status bits (CS0_PAGED and CS1_PAGED) should be used only for a page-related command like read or write. 0 = No Boundary: The current read/write transaction on NF_CEn[1] has not reached a page boundary. 1 = Boundary: The current read/write transaction on NF_CEn[1] has reached a page boundary.
8	CS0_CMDD	RW 0x0	NF_CSn[0,2] Command Done The NF_CSn[0,2] command-done (CS0_CMDD) bit is set when the execution of the command sent to the flash device(s) on NF_CSn[0,2] is completed successfully. CS0_CMDD is not set if an auto-status check after a program/erase command returns a failure. Command-done condition can be programmed to cause an interrupt by clearing the NF_CSn[0,2] command-done mask (CS0_CMDDM). Command done is set when the command has finished being sequenced to the DFI, but whether or not a particular command is finished depends on the specific command. A read type command is done when it has been sequenced but it is still required that the data buffer be emptied, so setting CS0_CMDD does not signify that the NFC is ready for the next operation. 0 = Incomplete: The command execution on NF_CEn[0] has not successfully completed. 1 = Complete: The command execution on NF_CEn[0] has successfully completed.

Copyright © 2015 Marvell

Table 293: NAND Controller Status Register (NDSR) (Continued)
Offset: 0x000D0014

Bit	Field	Type/InitVal	Description
7	CS1_CMDD	RW 0x0	NF_CEn[1,3] Command Done The NF_CEn[1,3] command-done (CS1_CMDD) bit is set when the execution of the command sent to the flash device(s) on NF_CEn[1,3] is completed successfully. CS1_CMDD is not set if an auto-status check after a program/erase command returns a failure. Command-done condition can be programmed to cause an interrupt by clearing the NF_CEn[1,3] command-done mask (CS1_CMDDM). Command done is set when the command has finished being sequenced to the DFI, but whether or not a particular command is finished depends on the specific command. A read type command is done when it has been sequenced but it is still required that the data buffer be emptied, so setting CS0_CMDD does not signify that the NFC is ready for the next operation. 0 = Incomplete: The command execution on NF_CEn[1] has not successfully completed. 1 = Complete: The command execution on NF_CEn[1] has successfully completed.
6	CS0_BBD	RW 0x0	NF_CSn[0,2] Bad Block Detect The NF_CSn[0,2] bad-block detect (CS0_BBD) status bit is set when the status read after a program/erase operation to flash device(s) interfaced using NF_CSn[0,2] returns a program/erase failure. A bad-block detect condition can be programmed to cause an interrupt by clearing the NF_CSn[0,2] bad-block detect interrupt mask (CS0_BBDM). When two 8-bit NAND flash devices are interfaced to the 16-bit NAND flash controller data bus (NDCR[DWIDTH_C] = 1 and NDCR[DWIDTH_M] = 0) using NF_CSn[0,2], a program/erase failure in any one of the two devices results in the CS0_BBD bit getting set. Because both flash devices are addressed at the same time, software must mark the corresponding blocks as non-valid. 0 = No Bad Block: No bad block is encountered while a write/erase on NF_CEn[0]. 1 = Bad Block: Bad block is encountered while a write/erase on NF_CEn[0].
5	CS1_BBD	RW 0x0	NF_CEn[1,3] Bad Block Detect The NF_CEn[1,3] bad-block detect (CS1_BBD) status bit is set when the status read after a program/erase operation to flash device(s) interfaced using NF_CEn[1,3] returns a program/erase failure. A bad-block detect condition can be programmed to cause an interrupt by clearing the NF_CEn[1,3] bad block detect interrupt mask (CS1_BBDM). When two 8-bit NAND flash devices are interfaced to the 16-bit NAND flash controller data bus (NDCR[DWIDTH_C]= 1 and NDCR[DWIDTH_M] = 0) using NF_CEn[1,3], a program/erase failure in any one of the two devices results in CS1_BBD bit getting set. Because both flash devices are addressed at the same time, software can mark the corresponding blocks as non-valid. 0 = No Bad Block: No bad block is encountered while a write/erase on NF_CEn[1]. 1 = Bad Block: Bad block is encountered while a write/erase on NF_CEn[1].

Copyright © 2015 Marvell

Table 293: NAND Controller Status Register (NDSR) (Continued)
Offset: 0x000D0014

Bit	Field	Type/InitVal	Description
4	UNCERR	RW 0x0	Uncorrectable Error The uncorrectable error (UNCERR) status bit is set when an uncorrectable error is detected in any of the read data streams. Uncorrectable error condition can be programmed to cause an interrupt by clearing the uncorrectable error mask (UNCERRM). For NDECCCTRL.BCH_EN == 0 && NDCR.ECC_EN == 1 (i.e. Hamming ECC is enabled) an uncorrectable error is the detection of two errors in a syndrome range. Hamming codes are guaranteed to detect all possible double bit ECC error over their syndrome range. For NDECCCTRL.BCH_EN == 1&& NDCR.ECC_EN == 1 (i.e. BCH ECC is enabled) an uncorrectable error is the detection of more errors than be corrected. BCH codes are not guaranteed to detect all possible error cases in excess of their correction ability, and it is possible (although highly unlikely) for the BCH engine to report a false correction success. When an uncorrectable error is detected, the bad-block detect bits (CS0_BBD, CS1_BBD) are not set and NAND Control Bad Block registers (NDBBRx) are not updated. Software must take corrective action in this scenario. Software could read NDCBx registers to get the block address that returned erroneous data and mark this block as invalid. 0 = No error: No uncorrectable error is encountered in any of the page read data streams 1 = Error: Uncorrectable error is encountered in one of the page read data streams
3	CORERR	RW 0x0	Correctable Error The CORERR status bit is set when a correctable error is detected in any of the read data streams. A correctable error condition can be programmed to cause an interrupt by clearing the correctable error mask (CORERRM). For NDECCCTRL.BCH_EN == 0 && NDCR.ECC_EN == 1 (i.e. Hamming ECC is enabled) a correctable error is the detection of one error in a syndrome range. For NDECCCTRL.BCH_EN == 1 && NDCR.ECC_EN == 1 (i.e. BCH ECC is enabled) a correctable error is the detection of greater than NDECCCTRL.ECC_THRESH errors and less than or equal to the maximum number of errors the BCH can correct. **NOTE:** If CORERR is set, ERR_CNT will be set to 0x1f. 0 = No Error: No correctable error is encountered in any of the page read data streams. 1 = Error: Correctable error is encountered in one of the page read data streams.
2	WRDREQ	RW 0x0	Write Data Request The write-data-request (WRDREQ) status bit is set when a write command is loaded into the NAND flash controller command buffer and data buffer is empty. The write data request status bit can be programmed to cause an interrupt by clearing the write-data-request mask (WRDREQM). 0 = No Write: No write to the data buffer is required. 1 = Write: Current command is a page write and data buffer has not been loaded.

Copyright © 2015 Marvell

Table 293: NAND Controller Status Register (NDSR) (Continued)
Offset: 0x000D0014

Bit	Field	Type/InitVal	Description
1	RDDREQ	RW 0x0	Read Data Request The read-data-request (RDDREQ) status bit is set when the data buffer is loaded with a complete page (including spare data if SPARE_EN is set) of read data for a read operation. For read ID and read-status operations, RDDREQ is set when the data buffer is loaded with the requested number of read data bytes as specified in the command. When ECC is enabled for Hamming, RDDREQ is set once per entire data buffer transfer. When BCH ECC is enabled, RDDREQ is set once per 32 bytes of read data transfer. The processor must check that this bit is set after each 32 byte read from NDDB. 0 = No read: No read from the data buffer is required. 1 = Read: Data buffer has read data available.
0	WRCMDREQ	RW 0x0	Write Command Request The write-command-request (WRCMDREQ) status bit is set when one of the following conditions is met, indicating a request to write a command to the NAND flash controller command buffer. NDCR[ND_RUN] is set and the command buffer is empty. The Ready/Busy_ (R/B_) input is asserted low after the transfer of any command except read ID and read status, with 3 (next command) it set, to the addressed NAND flash device. After the completion of read-ID or read-status commands with next command (NC) bit set. Write command request can be programmed to cause an interrupt by clearing the write-command request mask (WRCMDREQM). 0 = No Write Required: No write to the command buffer is required. 1 = Write Required: New Command needs to be written to the command Buffer.

Table 294: NAND Controller Page Count Register (NDPCR)
Offset: 0x000D0018

Bit	Field	Type/InitVal	Description
This register can be read to determine the completed number of pages for multi-page program/read operations. Writes to NDPCR are ignored. This is a read-only register. Ignore reads from reserved bits.			
31:24	Reserved	RSVD 0x0	Reserved
23:16	PG_CNT_1	RO 0x0	Page count for device interfaced using NF_CSn[1,3] The page count for a flash device on NF_CSn[1,3] (PG_CNT_1) indicates the number of pages completed (programmed or read) for a given NAND flash-controller command. NDCB0[AUTO_RS] must be set for a multi-page program or multi-page read command. The page count gets reset when a command execution is finished. PG_CNT_1 is incremented after each page of data is programmed or read. However, if the status check after programming a page indicates a failure, PG_CNT_1 is not incremented and software can read this value to find out the last page programmed correctly and then set NDCR[CLR_PG_CNT] to clear the page count values.
15:8	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 294: NAND Controller Page Count Register (NDPCR) (Continued)
Offset: 0x000D0018

Bit	Field	Type/InitVal	Description
7:0	PG_CNT_0	RO 0x0	Page count for device interfaced using NF_CSn[0,2] Page count for Flash Device on NF_CSn[0,2] (PG_CNT_0) indicates the number of pages completed (programmed or read) for a given NAND flash controller command. NDCB0[AUTO_RS] must be set for a multi-page program or multi-page read command. The page count gets reset when a command execution is finished. PG_CNT_0 is incremented after each page of data is programmed or read. However, if the status check after programming a page indicates a failure, PG_CNT_0 is not incremented and software can read this value to find out the last page programmed correctly and then set NDCR[CLR_PG_CNT] to clear the page count values.

Table 295: NAND Controller Bad Block Registers (NDBBRx)
Offset: 0x000D001C

Bit	Field	Type/InitVal	Description
This register holds the bad-block information when a bad block is detected. NDBBR0 holds the bad-block information for a bad-block detected for a device interfaced using NF_CEn[0] (NDSR[CS0_BBD] is set). Similarly, NDBBR1 holds the bad-block information for a device interfaced using NF_CEn[1] (NDSR[CS1_BBD] set). These are read-only registers. Ignore reads from reserved bits.			
31:0	Bad_Block_Information	RO 0x0	For a program command, bad block information contained in these registers consists of ADDR5, ADDR4, ADDR3, and ADDR2 for the command that resulted in a bad block detection. ADDR5 occupies the most significant byte and ADDR2 the least significant byte in NDBDRx. For an erase command bad block information contained in these registers consists of ADDR4, ADDR3, ADDR2, and ADDR1 for the command that resulted in a bad block detection. ADDR4 occupies the most significant byte and ADDR1 the least significant byte in NDBDRx. When a bad block is detected (the CS0_BBD or CS1_BBD bits in NDSR gets set), software reads the corresponding NDBDRx registers, uses the relevant portions of the bad-block information to get the bad-block address, and mark these blocks addresses as non-valid. Writes to the NDBBRx registers are ignored.

Table 296: NAND ECC Control (NDECCCTRL) Register
Offset: 0x000D0028

Bit	Field	Type/InitVal	Description
31:15	Reserved	RSVD 0x0	Reserved
14:7	ECC_SPARE	RO 0x0	ECC Bytes used in Spare Area This read-only value determines the number of bytes in the spare area that are dedicated to ECC storage by the hardware. The number of bytes is determined by NDCR.ECC_EN and BCH_EN. To compute the number of available spare bytes available to the programmer, take the spare size (16 bytes for a small block and 64 bytes for a large block) and subtract the value in ECC_SPARE. Note that this value is per chunk, so for a 4K page size that is read and written in two chunks this value specifies the ECC storage per chunk, not in total.

Copyright © 2015 Marvell

Table 296: NAND ECC Control (NDECCCTRL) Register (Continued)
Offset: 0x000D0028

Bit	Field	Type/InitVal	Description
6:1	ECC_THRESH	RW 0x0	ECC Warning Threshold ECC threshold. When BCH_EN is true and ECC_EN in the NDCR is true, and the number of corrected errors exceeds this value, the NDSR.CORERR status is set and NDSR.ECC_CNT contains the number of corrected error bits. Currently, with a maximum of 16 errors correctable, values greater than 15 have the effect of disabling CORERR status. **NOTE:** If it is desired to always see all correctable errors, this field should be set to 0.
0	BCH_EN	RW 0x0	BCH Enable When ECC_EN in the NDCR is true, reads from a page will perform ECC to the following degrees of correctability: **NOTE:** There is an interaction between the ganging configuration and the ECC choice in that ganging 512 byte page size devices is only supported with the Hamming ECC engine and the ganging of 2048 (or 4096) byte page size devices is only supported with the BCH ECC engine. 0 = 2/8-bits: 2 bits per 512 byte page and 8 bits per 2048 byte page, distributed as one error in every other bit per 512 byte sector using a Hamming algorithm. 1 = 16-bits: 16 bits per 2048 byte page (plus spare area, if enabled).

Table 297: NAND Busy Length Count (NDBZCNT) Register
Offset: 0x000D002C

Bit	Field	Type/InitVal	Description
31:16	ND_RBnCNT1	RO 0x0	RBn Busy Count 1 This field is cleared to zero on the 1 -> 0 transition of NF_RBn[1] and while NF_RBn[1] is low, the field counts basic NFC controller clocks. When NF_RBn[1] has a 0 -> 1 transition, the count freezes.
15:0	ND_RBnCNT0	RO 0x0	RBn Busy Count 0 This field is cleared to zero on the 1 -> 0 transition of NF_RBn[1] and while NF_RBn[0] is low, the field counts basic NFC controller clocks. When NF_RBn[0] has a 0 -> 1 transition, the count freezes.

Copyright © 2015 Marvell

Table 298: NAND Controller Data Buffer (NDDB) Register
Offset: 0x000D0040

Bit	Field	Type/InitVal	Description
NAND controller data buffer (NDDB) is the read/write port of the NAND controller data buffer. NDDB is the source for read-data, and the destination for write-data. Only four byte writes and four byte reads are supported for software accesses to NDDB, mainly because the application processor cannot perform an 8-byte read. Read/write operations to NDDB must occur in the following order: The first write corresponds to the lower four bytes of the data buffer and the following operation corresponds to the upper four bytes. This register behaves as though it were a FIFO to the actual 2176 byte data buffer. The buffer contains 2KB for the “main” data area plus an additional 128 bytes to accommodate the currently defined spare size as well as allow for increases in the future. When ECC is enabled for Hamming, NDSR.RDDREQ is set once per entire data buffer transfer. When BCH ECC is enabled, NDSR.RDDREQ is set once per 32 bytes of data read transfer. The processor must check that this bit is set after each 32 byte read from NDDB. This is a read/write register. Ignore reads from reserved bits. Write 0b0 to reserved bits.			
31:0	NAND_flash_data	RW 0x0	Holds the write/read data

Table 299: NAND Controller Command Buffer 0 (NDCB0) Register
Offset: 0x000D0048

Bit	Field	Type/InitVal	Description
The NAND controller-command buffer 0 (NDCB0) holds the commands (CMD1, CMD2) to be sent to the NAND flash device and command-control (CMD_CTRL) information. This register determines the overall semantic structure of the command sequence. This is a read/write register. Ignore reads from reserved bits. Write 0b0 to reserved bits.			
31:29	CMD_XTYPE	RW 0x0	Command Extended Type This field extends the command type semantic field and is defined only for read and write semantics. The read semantic is divided into three sections, command dispatch, data transfer (after ready is high) and DMA transfer. If the third section is disabled, the operation being performed is referred to as “read sniffing”. Here the purpose is to read a page and compute the number of ECC correction bits to make sure that the total correction is within the safe range of the NAND management algorithm. The write semantic is divided into three sections, initial command dispatch, data transfer and command issue. The reason for this extended field is to enable larger block support. All blocks larger than 2KB will composed of 2KB chunks, and to simplify the controller logic, these chunks are moved to and from the NAND as a separate command semantic. Read Semantics: 0b000 = Monolithic Read 0b001 = Last Naked Read 0b010 = Illegal 0b011 = Illegal 0b100 = Read 0b101 = Naked Read 0b110 = Command dispatch 0b111 = Illegal Write Semantics: 0b000 = Monolithic Write 0b001 = Naked Write with final command 0b010 = Illegal 0b011 = Final Command 0b100 = Command dispatch with write 0b101 = Naked Write 0b110 = Command dispatch 0b111 = Illegal
28	LEN_OVRD	RW 0x0	Length Override Length override is used for non-standard length operations such as the ONFI parameter page read. 0 = Three Writes: Use the lengths defined by NDCR.PAGE_SZ for the operation. A command block consists of three writes and the NDLENCNT value in NDCB3 may not be written. 1 = Four Writes: Use the length defined in NDCB3.NDLENCNT for the operation, regardless of the programming of NDCR.PAGE_SZ. A command block consists of four writes and the NDCB3 register must be written.

Copyright © 2015 Marvell

Table 299: NAND Controller Command Buffer 0 (NDCB0) Register (Continued)
Offset: 0x000D0048

Bit	Field	Type/InitVal	Description
27	RDY_BYP	RW 0x0	Ready Bypass The purpose for this function is to enable a real-time schedule of operations to the two chip selects where operations on one NAND can continue while operations on the other NAND are in progress. This is useful to enable striping where multiple identical operations are scheduled to the two chip selects. This bit is in addition to the capability mentioned in Command Sequence and Parallel Execution register, and allows virtually any operation pair to be overlapped. 0 = Wait: The current command will wait for the tR timeout as specified in NDTR1CS0.tR_NFC and NDTR1CS0.WAIT_MODE for the indicated CSEL to be true before proceeding according to the bypass rules. (In other words, the current command will wait for the selected RDY to be true.) 1 = Continue: The current command will continue to the next command (if there is one as defined by the NC bit), regardless of the state of RDY.
26	ST_ROW_EN	RW 0x0	Status Row Address Enable For ONFI NAND and other NAND parts, the status command may be performed on a per logical unit basis. The logical unit is addressed by the row address defined in the read, program or erase command. If this bit is set, both the automatic status cycle at the end of each read or write or any “manual” status read will supply the row address where the row address is defined by the ADDR_CYC bits and the RA_START bit in the NDCR. For read and program commands the ADDR_CYC defines the total number of address cycles and the RA_START bit defines which address bytes form the row address. For the erase command, the ADDR_CYC itself defines the number of row address cycles and the RA_START bit is ignored. It is expected that this bit is set for status commands and not for AUTO_RS. If multiple LUN program and erase commands are being issued, it will be necessary to issue LUN status commands with this bit enabled to determine precisely which LUN caused the failure.
25	AUTO_RS	RW 0x0	Auto Read Status When set, the auto-read status bit (AUTO_RS) enables the automatic checking of the program/erase status by issuing a read status command. The bit pattern to be used for the status command is defined in NDCB2. When AUTO_RS is clear, no automatic status check is performed. When automatic status check is performed, status-read data is not written into the data buffer. The status-read data is checked by the NAND flash controller for the success of program/erase operation, and in case of failure, the appropriate bad-block-detected bits are updated in NDSR, the address of the bad block is saved in the NDBBRLx registers. Set AUTO_RS for program and erase commands only. Setting AUTO_RS for commands other than program or erase may result in incorrect operation by the NAND flash controller. AUTO_RS must be set for multipage program (when NDCB2[Page_Count] is non-zero) to ensure that a read-status command is issued after every page program to verify the success of the program operation. 0 = No Automatic Read: No automatic read status command execution after program/erase. 1 = Automatic Read: Automatic read status command execution after program/erase.

Copyright © 2015 Marvell

Table 299: NAND Controller Command Buffer 0 (NDCB0) Register (Continued)
Offset: 0x000D0048

Bit	Field	Type/InitVal	Description
24	CSEL	RW 0x0	CS Select The CS-select bit selects the chip-select signal (NF_CSn[0] or NF_CSn[1]) to be activated for the command execution. 0 = NF_CSn[0]: NF_CSn[0] is asserted for the access. 1 = NF_CSn[1]: NF_CSn[1] is asserted for the access.
23:21	CMD_TYPE	RW 0x0	Command Type Command type (CMD_TYPE) defines the type of NAND flash command represented by CMD1 (and CMD2 if for a double byte command). **NOTE:** No data transfer occurs when erase or reset commands are executed. The Naked Command issues CMD1 as a command cycle and continues. The reason for this semantic is to enable the creation of “outlier” command semantics such as “Read Unique ID”. The Naked Address issues ADDDR1 as a command cycle and continues. The reason for this semantic is to enable the creation of “outlier” command semantics such as “Read Unique ID”. The format of the data provided by a Read ID and Status Read is defined by the NAND manufacturer. Note that when devices are ganged, both the status and ID information is replicated on each half of each 16-bit word. Note also that for status, the value for each part in the gang is device dependent. 0 = Read 1 = Program (Write) 2 = Erase 3 = Read ID 4 = Status Read 5 = Reset 6 = Naked Command 7 = Naked Address
20	NC	RW 0x0	Next Command The next-command (NC) bit (if set) indicates the presence of another valid command following the current command. The last command of a command sequence must have the NC bit clear. 0 = No Valid Command: No valid command following the current command. 1 = Valid Command: Valid command following the current command.
19	DBC	RW 0x0	Double Byte Command The double-byte command (DBC) bit (if set) indicates that the current command involves the transfer of two commands to the NAND flash. DBC must be clear for a single-byte command. 0 = Single-Byte: Current command is a single-byte command. 1 = Double-Byte: Current command is a double-byte command.
18:16	ADDR_CYC	RW 0x0	Number of Address Cycles Address cycles (ADDR_CYC) specifies the number of address cycles in which the NAND flash is to be addressed. Valid values for address cycles are 1 to 5 for NAND commands Program, Read, Erase and Read-Id. For example, if ADDR_CYC value is programmed as 4, ADDR1, ADDR2, ADDR3, and ADDR4 are sent to the flash device in successive address cycles, while ADDR5 is ignored. Valid values for address cycles for Reset and Read-status commands is 0. If the SW programs a non-zero value for these commands the NAND controller will simply ignore it.

Copyright © 2015 Marvell

Table 299: NAND Controller Command Buffer 0 (NDCB0) Register (Continued)
Offset: 0x000D0048

Bit	Field	Type/InitVal	Description
15:8	CMD2	RW 0x0	Second Command The second command (CMD2) contains the second byte of command sent to NAND flash after CMD1 and address in a double-byte command.
7:0	CMD1	RW 0x0	First command The first command (CMD1) contains the first byte command sent to the NAND flash when the command execution begins.

Table 300: NAND Controller Command Buffer 1 (NDCB1) Register
Offset: 0x000D004C

Bit	Field	Type/InitVal	Description
This register contains the addresses to be sent to the NAND flash in multiple cycles. In NAND-flash mode, I/O<7:0> pins are used to send ADDRx<7:0>, in as many cycles as defined by NDCB0[ADDR_CYC]. The ADDR5 field in NAND controller command buffer 2 (NDCB2) is used to specify the fifth cycle of address for devices supporting five addressing cycles. Contents of the ADDRx fields are sent out sequentially by the NAND controller during the addressing phase. Program these fields with address values based on the cycle-by-cycle addressing specified by the NAND-flash device. When programming an erase command the contents of ADDR1 must be replicated in ADDR5 in order to save the full address during a bad block detect since only fields ADDR2 through ADDR5 are saved. When programming a manual read status command the user must write the ADDRx fields with the same address as the previous command in order to save a valid address to NDBDRx during a bad block detect. This is a read-only register. Ignore reads from reserved bits.			
31:24	ADDR4	RO 0x0	Address sent out to the flash device on the fourth addressing cycle.
23:16	ADDR3	RO 0x0	Address sent out to the flash device on the third addressing cycle.
15:8	ADDR2	RO 0x0	Address sent out to the flash device on the second addressing cycle.
7:0	ADDR1	RO 0x0	Address sent out to the flash device on the first addressing cycle.

Table 301: NAND Controller Command Buffer 2 (NDCB2) Register
Offset: 0x000D0050

Bit	Field	Type/InitVal	Description
This register holds the address (ADDR5) for the fifth addressing cycle in NAND-flash mode and the page count for the command. This is a read-only register. Ignore reads from reserved bits.			
31:24	ST_MASK	RO 0x0	Status Mask This is the mask applied to the status field to determine failure. This field value is ANDed with the status word received from the NAND device and the resulting byte is bitwise ORed over the [7:0] field. If the result is true (i.e. any bit allowed by ST_MASK is set), the status is indicating failure. A Status Mask of 0x00 is singular in that it defines a mask of 0x01 instead. If software issues 0x00 to this field, the value actually loaded will be 0x01. Receiving bad status sets one of the CSx_BBD bits in NDSR (depending on the active chip select) when AUTO_RS is set. Generally, this field should only be set to a non-standard value when performing non-standard status commands as a stands-alone command.

Copyright © 2015 Marvell

Table 301: NAND Controller Command Buffer 2 (NDCB2) Register (Continued)
Offset: 0x000D0050

Bit	Field	Type/InitVal	Description
23:16	ST_CMD	RO 0x0	Status Command The default status command is 0x70 but ONFI and other devices allow logical unit status checks. This field allows the non-standard status command to be defined for both status commands as well as auto status reads. A Status Command of 0x00 is singular in that it defines a status command of 0x70 instead. If software issues 0x00 to this field, the value actually loaded will be 0x70. While it is possible to use LUN status for AUTO_RS, there is only one ST_CMD, ST_MASK and ST_ROW_EN value, but there are two AUTO_RS for overlapped operations. This means that one cannot have different values for these fields on any pair of operations that can potentially overlap. For this reason, the use of LUN status during multiple chained operations is discouraged. Instead, the NFC should perform simple status operations, which will perform the “OR” of all logical operations in the NAND and when a bad block is detected, software can issue LUN status commands to determine exactly which LUN operation actually failed.
15:8	Page_Count	RO 0x0	Page Count Page_Count specifies the number of pages of data to be transferred for a program or read command. Set this field to zero for all commands except multipage program/read commands (commands where more than one page is programmed or read), in which case Page_Count + 1 specifies the number of pages serviced. Value from 0 to 255. Specifies the number of pages of data to be transferred for the current command. This field must not be set to a larger value than allowed by NDCR.PG_PER_BLK. Number of pages to be transferred = Page_Count +1 **NOTE:** While multi-page transfers reduce the number of command cycle that need to be programmed into NDCBx, they do not necessarily improve throughput because each operation to the NAND is treated separately. (Refer to the description of <SEQ_DIS> in Table 381 on page 299). To truly optimize throughput requires cache instructions and/or LUN commands and a carefully constructed real time schedule.
7:0	ADDR5	RO 0x0	Address sent out to the flash device on the fifth addressing cycle. 0 = CS0_1: CS0 or CS1 selected according to the value NDCB0 <CSEL> bit[24]. 1 = CS2_3: CS2 or CS3 selected according to the value NDCB0 <CSEL> bit[24].

Copyright © 2015 Marvell

Table 302: NAND Controller Command Buffer 3 (NDCB3) Register
Offset: 0x000D0054

Bit	Field	Type/InitVal	Description
For increased flexibility and to support ONFI, this register allows arbitrary length read data transfers. For any transfer that uses this arbitrary length count, NDCR.SPARE_EN must be disabled or the result will be unpredictable. If NDCR.ECC_EN is enabled, ECC is applied as described elsewhere. The register address is after the NDCBx registers and the register is written as an optional extension of writing a command block to NDCB0. If NDCB0.LEN_OVRD is false and NDCB0.ADDR_CYC < 6, this register must not be written and a fourth write to NDCB0 is ignored. If NDCB0.LEN_OVRD is true or NDCB0.ADDR_CYC >= 6, this register must be written, even if it is not changing value.			
31:24	ADDR7	RO 0x0	Address sent out to the flash device on the seventh addressing cycle.
23:16	ADDR6	RO 0x0	Address sent out to the flash device on the sixth addressing cycle.
15:0	NDLENCNT	RO 0x0	When the arbitrary length count is selected, this value is used to count the data cycles for any and all selected operations. As a practical matter, this value should not exceed the 2176 byte buffer size or data will overflow the data FIFO. When ECC is enabled and BCH is selected, this value must be a multiple of 32 bytes. **NOTE:** The value 0 is defined to be valid, and when this field is zero, no data transfer is performed.

Copyright © 2015 Marvell

A.5 Device Bus Controller Registers

The following table provides a summarized list of all registers that belong to the Device Bus Controller, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 303: Summary Map Table for the Device Bus Controller Registers

Register Name	Offset	Table and Page
Device Bus		
DEV_BOOTCSn Read Parameters Register	0x00010400	Table 304, p. 544
DEV_BOOTCSn Write Parameters Register	0x00010404	Table 305, p. 545
DEV_CSn[n] Read Parameters Register (n=0–3)	CS0: 0x00010408, CS1: 0x00010410, CS2: 0x00010418, CS3: 0x00010420	Table 306, p. 546
DEV_CSn[n] Write Parameters Register (n=0–3)	CS0: 0x0001040C, CS1: 0x00010414, CS2: 0x0001041C, CS3: 0x00010424	Table 307, p. 546
Device Bus Interface Control Register	0x000104C0	Table 308, p. 547
Device Bus Error Address Register	0x000104C4	Table 309, p. 548
Device Bus Synchronous Control Register	0x000104C8	Table 310, p. 548
Device Bus Interrupt Cause Register	0x000104D0	Table 311, p. 550
Device Bus Interrupt Mask Register	0x000104D4	Table 312, p. 550
Device Bus Miscellaneous Timing Parameters Register	0x000104D8	Table 313, p. 550

A.5.1 Device Bus

Table 304: DEV_BOOTCSn Read Parameters Register
Offset: 0x00010400

Bit	Field	Type/InitVal	Description
31:30	DevWidth	RW 0x0	Boot Device Width 0 = 8-bit 1 = 16-bit 2 = Reserved: Reserved 3 = Reserved
29:28	Reserved	RSVD 0x0	Reserved
27:23	RdHold	RW 0x0	Defines the number of TCLK cycles in read accesses between de-assertion of DEV_OEn and de-assertion of DEV_BOOTCSn. **NOTE:** Set <RdHold> to a value smaller than <TurnOff>.- 2.

Document Classification: Proprietary Information

Table 304: DEV_BOOTCSn Read Parameters Register (Continued)
Offset: 0x00010400

Bit	Field	Type/InitVal	Description
22:17	Acc2Next	RW 0x3f	Defines the number of TCLK cycles between the cycle that samples data N and the cycle that samples data N+1 (in burst accesses). Minimal value = 0x2
16:12	RdSetup	RW 0x0	Defines the number of TCLK cycles in read accesses between the end of the address phase and the assertion of DEV_OEn. End of address phase occurs 1 cycle after DEV_ALE[0] negation.
11:6	Acc2First	RW 0x3f	Defines the number of TCLK cycles in a read access between the negation of DEV_ALE[0] and the cycle containing the first data sampled by the device. Number of cycles = <Acc2First> - 2 * <Addr2Ale> - 1. Minimal value = 2 * <Addr2Ale> + 2 + <RdSetup> + 2.
5:0	TurnOff	RW 0x0f	Defines the minimal gap guaranteed between 2 consecutive device accesses. The number of TCLK cycles between de-assertion of DEV_BOOTCSn and the beginning of the next address phase will be at least <TurnOff> - <RdHold>. Minimal value for <TurnOff> is <RdHold>+2.

Table 305: DEV_BOOTCSn Write Parameters Register
Offset: 0x00010404

Bit	Field	Type/InitVal	Description
31:25	Reserved	RSVD 0x0	Reserved
24	DevSyncEnable	RW 0x0	Synchronous Device Enable 0 = False 1 = True
23:22	Reserved	RSVD 0x0	Reserved
21:16	WrHigh	RW 0xf	The number of TCLK cycles in a burst write access where DEV_WEn is kept de-asserted. Minimal value = 0x1.
15:14	Reserved	RSVD 0x0	Reserved
13:8	WrLow	RW 0xf	The number of TCLK cycles in a write access where DEV_WEn is kept active. Minimal value = 0x1.
7:6	Reserved	RSVD 0x0	Reserved
5:0	ALE2Wr	RW 0xf	Defines the number of TCLK cycles in a write access between DEV_ALE[0] negation and the assertion of DEV_WEn. Number of cycles = <ALE2Wr> - 2 * <Addr2ALE> - 1. Minimal value = 2 * <Addr2ALE> + 2.

Copyright © 2015 Marvell

Table 306: DEV_CSn[n] Read Parameters Register (n=0–3)
Offset: CS0: 0x00010408, CS1: 0x00010410, CS2: 0x00010418, CS3: 0x00010420
Offset Formula: 0x00010408+n * 8: where n (0-3) represents CS

Bit	Field	Type/InitVal	Description
31:30	DevWidth	RW 0x0	Device Width 0 = 8-bit 1 = 16-bit 2 = Reserved 3 = Reserved
29:28	Reserved	RSVD 0x0	Reserved
27:23	RdHold	RW 0x0	Defined the number of TCLK cycles in read accesses between de-assertion of DEV_OEn and de-assertion of DEV_CSn. Minimum value=0. Maximal value=<TurnOff> - 2.
22:17	Acc2Next	RW 0x1F	Defines the number of TCLK cycles between the cycle that samples data N and the cycle that samples data N+1 (in burst accesses). Minimal value = 0x2.
16:12	RdSetup	RW 0x0	Defined the number of TCLK cycles in read accesses between the end of the address phase and the assertion of DEV_OEn. The end of address phase occurs 1 cycle after DEV_ALE[0] negation.
11:6	Acc2First	RW 0x1F	Defines the number of TCLK cycles in a read access between the negation of DEV_ALE[0] and the cycle containing the first data sampled by the device. Number of cycles = Acc2First - 2*Addr2Ale - 1. Minimal value = 2 * Addr2Ale + 2 + RdSetup + 2.
5:0	TurnOff	RW 0x0f	Defines the minimal gap guaranteed between 2 consecutive device accesses. The number of TCLK cycles between de-assertion of DEV_CSn and the beginning of the next address phase will be at least <TurnOff> - <RdHold>. Minimal value for <TurnOff> is <RdHold>+2.

Table 307: DEV_CSn[n] Write Parameters Register (n=0–3)
Offset: CS0: 0x0001040C, CS1: 0x00010414, CS2: 0x0001041C, CS3: 0x00010424
Offset Formula: 0x0001040C+ 8 * n: where n (0-3) represents CS

Bit	Field	Type/InitVal	Description
31:25	Reserved	RSVD 0x0	Reserved
24	DevSyncEnable	RW 0x0	Synchronous Device Enable 0 = False 1 = True
23:22	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 307: DEV_CSn[n] Write Parameters Register (n=0–3) (Continued)
Offset: CS0: 0x0001040C, CS1: 0x00010414, CS2: 0x0001041C, CS3: 0x00010424
Offset Formula: 0x0001040C+ 8 * n: where n (0-3) represents CS

Bit	Field	Type/InitVal	Description
21:16	WrHigh	RW 0xf	The number of TCLK cycles in a burst write access that the DEV_WEn signal is kept de-asserted. Minimal value = 0x1.
15:14	Reserved	RSVD 0x0	Reserved
13:8	WrLow	RW 0xf	The number of TCLK cycles in a write access that the DEV_WEn signal is kept active. Minimal value = 0x1.
7:6	Reserved	RSVD 0x0	Reserved
5:0	ALE2Wr	RW 0xf	Defines the number of TCLK cycles in a write access between DEV_ALE[0] negation and the assertion of DEV_WEn. Number of cycles = ALE2Wr - 2*Addr2ALE - 1. Minimal value = 2*Addr2ALE + 2.

Table 308: Device Bus Interface Control Register
Offset: 0x000104C0

Bit	Field	Type/InitVal	Description
31:20	Reserved	RSVD 0x0	Reserved
19	ForceParityErr	RW 0x0	By setting this bit, the device controller internal buffer will act as if the memory is corrupted. Thus, upon the use of these buffers, DPErrInt will be set. This bit is a debug ability and is provided to support software development for implementing interrupt processing of an internal SRAM parity detection.
18:17	Addr2ALE	RW 0x2	Defines the number of TCLK cycles in which address is steady before ALE assertions. **NOTE:** Addr2ALE imposes a limitation of the timing parameters that exist for each CS, since Acc2First and ALE2First must be configured such that they are expired after address latching is done. 0 = 1cycles: Address will be steady 1 core clock cycle before ALE assertion 1 = 2cycles: Address will be steady 2 core clock cycle before ALE assertion 2 = 3cycles: Address will be steady 3 core clock cycle before ALE assertion 3 = Reserved: Invalid value
16	OEWE Shared	RW 0x0	Defines if OE and WE are latched at the first ALE cycle as A[15] and A[16]. This fact influences the OEn and WEn signals as following: 0 = OE and WE are not shared whenever CS is inactive OE and WE are inactive. 1 = OE and WE are shared with A[16:15]. Whenever CS is inactive and ALE[1:0] are high, OE and WE are inactive.

Table 308: Device Bus Interface Control Register (Continued)
Offset: 0x000104C0

Bit	Field	Type/InitVal	Description
15:0	Timeout	RW 0xFFFF	Timeout Timer Preset Value If the device access is not completed within the period of this preset value (due to a lack of READYn assertion), the Device controller completes the transaction as if READYn was asserted, and it asserts an interrupt. **NOTE:** If set to 0x0, the Device controller waits for READYn assertion forever. The minimum value for the timeout is 0x40, thus exceeding all timing parameters values.

Table 309: Device Bus Error Address Register
Offset: 0x000104C4

Bit	Field	Type/InitVal	Description
31:0	Addr	RO 0x0	Latched address upon DReadyErr event After the address is latched, no new address is latched until the register is read.

Table 310: Device Bus Synchronous Control Register
Offset: 0x000104C8

Bit	Field	Type/InitVal	Description
This register provides support for a synchronous device.			
31	DEV_CLKOUT_Load_Active	RO 0x0	When updating the DEV_CLKOUT Divider Value, this bit indicates that action has been completed, and DEV_CLKOUT is indeed driving the core clock with the requested division. Software must poll this bit after reloading a new DEV_CLKOUT _VAL. 0 = Load Non-active: DEV_CLKOUT divider is not updated. 1 = Load Active: DEV_CLKOUT divider is updated
30:26	Reserved	RSVD 0x0	Reserved
25	CS3 Ready Polarity	RW 0x0	See BootCS Ready Polarity 0 = Active Low 1 = Active High
24	CS3 Ready Ignore	RW 0x0	See BootCS Ready Ignore
23:21	Reserved	RSVD 0x0	Reserved
20	CS2 Ready Polarity	RW 0x0	See BootCS Polarity 0 = Active Low 1 = Active High
19	CS2 Ready Ignore	RW 0x0	See BootCS Ignore

Copyright © 2015 Marvell

Table 310: Device Bus Synchronous Control Register (Continued)
Offset: 0x000104C8

Bit	Field	Type/InitVal	Description
18:16	Reserved	RSVD 0x0	Reserved
15	CS1 Ready Polarity	RW 0x0	See BootCS Polarity 0 = Active Low 1 = Active High
14	CS1 Ready Ignore	RW 0x0	See BootCS Ignore
13:11	Reserved	RSVD 0x0	Reserved
10	CS0 Ready Polarity	RW 0x0	See BootCS Polarity 0 = Active Low 1 = Active High
9	CS0 Ready Ignore	RW 0x0	See BootCS Ignore
8:6	Reserved	RSVD 0x0	Reserved
5	BootCS Ready Polarity	RW 0x0	Defines the polarity of DEV_READYn when BootCS is accessed. 0 = Active Low 1 = Active High
4	BootCS Ready Ignore	RW 0x0	When set, DEV_READYn is ignored.
3:0	DEV_CLKOUT Divide Value	RW 0x1	Defines the DEV_CLKOUT frequency in terms of the core clock divider values. 0 = Disable: DEV_CLKOUT will not toggle. 2 = 1 to 2 3 = 1 to 3 4 = 1 to 4 5 = 1 to 5 6 = 1 to 6 7 = 1 to 7 8 = 1 to 8 9 = 1 to 9 10 = 1 to 10 11 = 1 to 11 12 = 1 to 12 13 = 1 to 13 14 = 1 to 14 15 = 1 to 15

Copyright © 2015 Marvell

Table 311: Device Bus Interrupt Cause Register
Offset: 0x000104D0

Bit	Field	Type/InitVal	Description
NOTE: All cause bits are clear only. They are set upon error condition cleared upon a value write of 0. Writing a value of 1 has no affect.			
31:2	Reserved	RSVD 0x0	Reserved
1	DRdyErr	RW0C 0x0	Ready Timer expired
0	RuDPErr	RW0C 0x0	Internal Device Controller SRAM's Parity Error.

Table 312: Device Bus Interrupt Mask Register
Offset: 0x000104D4

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x0	Reserved
1	MaskReadyErr	RW 0x0	Masks the assertion of an interrupt in case of a Ready Timeout event Mask only affects the assertion of interrupt pins. It does not affect the setting of bits in the Cause register.
0	MaskParityErr	RW 0x0	Masks the assertion of an interrupt in case of an internal SRAM parity error event. Mask only affects the assertion of interrupt pins. It does not affect the setting of bits in the Cause register.

Table 313: Device Bus Miscellaneous Timing Parameters Register
Offset: 0x000104D8

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x3FE000	Reserved
5	RdHoldADEn	RW 0x1	When set, DEV_A[2:0] remains stable between OEn de-assertion and CS de-assertion. 0 = False 1 = True
4	DevBootCSAssertDelayEn	RW 0x0	Defines if DEV ADDR is guaranteed to be stable 4 core clock cycles before BootCS assertion. Setting this bit requires both: - Changing RdSetup of the corresponding CS to be at least 0x4 . - Ale2Wr of the corresponding CS to be at least 2*(Addr2ALE+1) + 4. 0 = False 1 = True

Document Classification: Proprietary Information

Table 313: Device Bus Miscellaneous Timing Parameters Register (Continued)
Offset: 0x000104D8

Bit	Field	Type/InitVal	Description
3	DevCS3AssertDelayEn	RW 0x0	Defines if DEV ADDR is guaranteed to be stable 4 core clock cycles before CS3 assertion. Setting this bit requires both: - Changing RdSetup of the corresponding CS to be at least 0x4. - Ale2Wr of the corresponding CS to be at least 2*(Addr2ALE+1) + 4 . 0 = False: DEV_A[2:0] is set together with CS assertion. 1 = True: DEV_A[2:0] is set 4 cycles prior to CS assertion.
2	DevCS2AssertDelayEn	RW 0x0	Defines if DEV ADDR is guaranteed to be stable 4 core clock cycles before CS2 assertion. Setting this bit requires both: - Changing RdSetup of the corresponding CS to be at least 0x4. - Ale2Wr of the corresponding CS to be at least 2*(Addr2ALE+1) + 4 . 0 = False: DEV_A[2:0] is set together with CS assertion. 1 = True: DEV_A[2:0] is set 4 cycles prior to CS assertion.
1	DevCS1AssertDelayEn	RW 0x0	Defines if DEV ADDR is guaranteed to be stable 4 core clock cycles before CS1 assertion. Setting this bit requires both: - Changing RdSetup of the corresponding CS to be at least 0x4. - Ale2Wr of the corresponding CS to be at least 2*(Addr2ALE+1) + 4 . 0 = False: DEV_A[2:0] is set together with CS assertion. 1 = True: DEV_A[2:0] is set 4 cycles prior to CS assertion.
0	DevCS0AssertDelayEn	RW 0x0	Defines if DEV ADDR is guaranteed to be stable 4 core clock cycles before CS0 assertion. Setting this bit requires both: - Changing RdSetup of the corresponding CS to be at least 0x4. - Ale2Wr of the corresponding CS to be at least 2*(Addr2ALE+1) + 4 . 0 = False: DEV_A[2:0] is set together with CS assertion. 1 = True: DEV_A[2:0] is set 4 cycles prior to CS assertion.

Copyright © 2015 Marvell

A.6 Time Division Multiplexing Registers

The following table provides a summarized list of all registers that belong to the Time Division Multiplexing, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 314: Summary Map Table for the Time Division Multiplexing Registers

Register Name	Offset	Table and Page
TDM-to-Mbus Bridge		
TDM Interface Pin Multiplexing Register	0x000B4000	Table 315, p. 553
TDM-Mbus Configuration Register	0x000B4010	Table 316, p. 553
Window0 Control Register	0x000B4030	Table 317, p. 554
Window0 Base Register	0x000B4034	Table 318, p. 554
Window1 Control Register	0x000B4040	Table 319, p. 554
Window1 Base Register	0x000B4044	Table 320, p. 555
Window2 Control Register	0x000B4050	Table 321, p. 555
Window2 Base Register	0x000B4054	Table 322, p. 555
Window3 Control Register	0x000B4060	Table 323, p. 556
Window3 Base Register	0x000B4064	Table 324, p. 556
TDM-Mbus Configuration 1 Register	0x000B4070	Table 325, p. 556
TDM Control		
PCM Control Register	0x000B0000	Table 326, p. 557
Channel Time Slot Control Register	0x000B0004	Table 327, p. 559
Channel 0 Delay Control Register	0x000B0008	Table 328, p. 560
Channel 1 Delay Control Register	0x000B000C	Table 329, p. 560
Channel n Enable and Disable Register (n=0–1)	Channel0: 0x000B0010, Channel1: 0x000B0020	Table 330, p. 561
Channel n Buffer Ownership Register (n=0–1)	Channel0: 0x000B0014, Channel1: 0x000B0024	Table 331, p. 561
Channel 0 Transmit Data Start Address Register	0x000B0018	Table 332, p. 562
Channel 0 Receive Data Start Address Register	0x000B001C	Table 333, p. 562
Channel 1 Transmit Data Start Address Register	0x000B0028	Table 334, p. 562
Channel 1 Receive Data Start Address Register	0x000B002C	Table 335, p. 562
Channel n Total Sample Count Register (n=0–1)	Channel0: 0x000B0030, Channel1: 0x000B0034	Table 336, p. 563
Number of Time Slots Register	0x000B0038	Table 337, p. 563
TDM PCM Clock Rate Divider Register	0x000B003C	Table 338, p. 564
Interrupt Event Mask Register	0x000B0040	Table 339, p. 564
Interrupt Status Mask Register	0x000B0048	Table 340, p. 564
Interrupt Reset Selection Register	0x000B004C	Table 341, p. 565
Interrupt Status Register	0x000B0050	Table 342, p. 565
Dummy Data for Dummy RX Write Register	0x000B0054	Table 343, p. 566
Miscellaneous Control Register	0x000B0058	Table 344, p. 566

Copyright © 2015 Marvell

Table 314: Summary Map Table for the Time Division Multiplexing Registers (Continued)

Register Name	Offset	Table and Page
Channel n Transmit Data Current Address Register (n=0–1)	Channel0: 0x000B0060, Channel1: 0x000B0068	Table 345, p. 567
Channel n Receive Data Current Address Register (n=0–1)	Channel0: 0x000B0064, Channel1: 0x000B006C	Table 346, p. 567
Current Time Slot Register	0x000B0070	Table 347, p. 567
TDM Revision Register	0x000B0074	Table 348, p. 567
TDM Channel n Debug Register (n=0–1)	Channel0: 0x000B0078, Channel1: 0x000B007C	Table 349, p. 567
TDM DMA Abort Register 1	0x000B0080	Table 350, p. 568
TDM DMA Abort Register 2	0x000B0084	Table 351, p. 568
TDM Channel 0 Wideband Delay Control Register	0x000B0088	Table 352, p. 569
TDM Channel 1 Wideband Delay Control Register	0x000B008C	Table 353, p. 569

A.6.1 TDM-to-Mbus Bridge

Table 315: TDM Interface Pin Multiplexing Register
Offset: 0x000B4000

Bit	Field	Type/InitVal	Description
.			
31:14	Reserved	RSVD 0x1	Reserved
13:1	Reserved	RSVD 0x0	Reserved
0	TDMOutEn	RW 0x1	When this bit is cleared, TDM output pins and the codec reset output signal are enabled.

Table 316: TDM-Mbus Configuration Register
Offset: 0x000B4010

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	TimeoutEn	RW 0x1	Crossbar Arbiter Timer Enable
15:8	Reserved	RSVD 0x0	Reserved
7:0	Timeout	RW 0xFF	Crossbar Arbiter Timeout Preset Value

Copyright © 2015 Marvell

Table 317: Window0 Control Register
Offset: 0x000B4030

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1's followed by sequence of 0's. The number of 1's specifies the size of the window in 64-KByte granularity (e.g., a value of 0x00FF specifies 256 = 16 MByte). **NOTE:** A value of 0x0 specifies 64-KByte size.
15:8	Attr	RW 0x0E	Specifies the target interface attributes associated with this window. **NOTE:** See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. **NOTE:** See the "PCI Express Address Decoding" section. Do not configure this field to the SDRAM Controller.
3:1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x1	Window 0 Enable 0x0 = Window is disabled. 0x1 = Window is enabled.

Table 318: Window0 Base Register
Offset: 0x000B4034

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0000	Base Address Used with the <Size> field to set the address window size and location. Corresponds to transaction address[31:16].
15:0	Reserved	RSVD 0x0	Reserved

Table 319: Window1 Control Register
Offset: 0x000B4040

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size See "Window0 Control Register". .
15:8	Attr	RW 0x0D	Target specific attributes depending on the target interface. See the "Window0 Control Register". .
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See "Window0 Control Register".
3:1	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 319: Window1 Control Register (Continued)
Offset: 0x000B4040

Bit	Field	Type/InitVal	Description
0	WinEn	RW 0x1	Window1 Enable. See "Window0 Control Register". .

Table 320: Window1 Base Register
Offset: 0x000B4044

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x1000	Base Address See "Window0 Base Register".
15:0	Reserved	RSVD 0x0	Reserved

Table 321: Window2 Control Register
Offset: 0x000B4050

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size See "Window0 Control Register".
15:8	Attr	RW 0x0B	Target specific attributes depending on the target interface. See "Window0 Control Register".
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See "Window0 Control Register".
3:1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x1	Window2 Enable See "Window0 Control Register".

Table 322: Window2 Base Register
Offset: 0x000B4054

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x2000	Base Address See "Window0 Base Register".
15:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 323: Window3 Control Register
Offset: 0x000B4060

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size See "Window0 Control Register".
15:8	Attr	RW 0x07	Target specific attributes depending on the target interface. See "Window0 Control Register".
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See "Window0 Control Register".
3:1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x1	Window3 Enable See "Window0 Control Register".

Table 324: Window3 Base Register
Offset: 0x000B4064

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x3000	Base Address See "Window0 Base Register". .
15:0	Reserved	RSVD 0x0	Reserved

Table 325: TDM-Mbus Configuration 1 Register
Offset: 0x000B4070

Bit	Field	Type/InitVal	Description
31:3	Reserved	RW 0x0	Reserved
2	Reserved	RW 0x0	Always write 1 to this bit.
1	Reserved	RW 0x0	Reserved
0	PCM I/F S/W Reset	RW 0x0	PCM I/F S/W Reset. Notes: 1. If the device uses an internally generated PCM clock (the PCM Control register, offset 0xB0000, <MstrPclk> bit[0] is cleared), assertion of this bit must be performed at least 2 ms after the <MstrPclk> field is cleared. 2. This register must normally be used only at the configuration stage. 0 = PCMRSTnON: Assert PCM i/f S/W Reset 1 = PCMRSTnOFF: De-assert PCM i/f S/W Reset

Document Classification: Proprietary Information

A.6.2 TDM Control

Table 326: PCM Control Register
Offset: 0x000B0000

Bit	Field	Type/InitVal	Description
31	PerfBit	RW 0x1	This bit controls how many locations (1 location is 32-bit wide) are popped or pushed in or out of the FIFO interfacing to the GBUS. This might be used when the GBUS is busy, as it helps the TDM design to be more tolerant of GBUS latencies. 0 = One location: Always one location of FIFO are read or written. 1 = Two locations: Always two locations of FIFO are read or written.
30:18	Reserved	RSVD 0x0	Reserved
17	CH1WBand	RW 0x0	Channel 1 Wideband When using Wideband mode, <CH1DlyEn> in this register must be set to 1
16	CH0WBand	RW 0x0	Channel 0 Wideband When using Wideband mode, <CH0DlyEn> in this register must be set to 1.
15	RESERVED	RW 0x0	RESERVED. This bit must be 0.
14	CH1QualTyp	RW 0x0	Channel 1 Qualifier Type This controls the way the qualifier for channel 1 receive and transmit is going to stay active.
13	CH0QualTyp	RW 0x0	Channel 0 Qualifier Type This controls the way the qualifier for channel 0 receive and transmit is going to stay active.
12	QualPol	RW 0x0	Qualifier Reference Polarity is referred to as the clock edge where the qualifier is sampled asserted or de-asserted by the external CODEC. The design drives the qualifier at the opposite edge from what is sampled by external CODEC.
11	CH1QualEn	RW 0x0	Channel 1 Rx and Tx Qualifier Enable 0 = Disable Qualifier: Disable Qualifier for Channel 1 Rx and Tx External CODEC has time slot (or delay) programming registers to understand when to send and receive data. 1 = Enable Qualifier: Enable Qualifier for Channel 1 Rx and Tx There is one qualifier for receive and one for transmit. The qualifier will go active for transmit, or receive, depending on the time slot (or delay) programmed for the Channel 1 receive and transmit. <CH1_QUAL_TYP> controls the duration for which the qualifier for Channel 1 receive and transmit will stay active.

Copyright © 2015 Marvell

Table 326: PCM Control Register (Continued)
Offset: 0x000B0000

Bit	Field	Type/InitVal	Description
10	CH0QualEn	RW 0x0	Channel 0 Rx and Tx Qualifier Enable 0 = Disable Qualifier: Disable Qualifier for Channel 0 Rx and Tx. External CODEC has time slot (or delay) programming registers to understand when to send and receive data. 1 = Enable Qualifier: Enable Qualifier for Channel 0 Rx and Tx. There is one qualifier for receive and one for transmit. The qualifier will go active for transmit, or receive, depending on the time slot (or delay) programmed for the Channel 0 receive and transmit. <CH0_QUAL_TYP> controls the duration for which the qualifier for Channel 0 receive and transmit will stay active.
9	CH1DlyEn	RW 0x0	Channel 1 Delay Control Enable 0 = Disable: Disable the use of <CH1_RX_DLY> and <CH1_TX_DLY> field in the Channel 1 Delay Control Register. The value in <CH1_TX_TSLOT> and <CH1_RX_TSLOT> field in the Channel Time Slot Control Register will be used instead for receive and transmit for channel 1 data. 1 = Enable: Enable the use of Channel 1 delay control <CH1_RX_DLY> and <CH1_TX_DLY> field in the Channel 1 Delay Control Register. The value in <CH1_TX_TSLOT> and <CH1_RX_TSLOT> field in the Channel Time Slot Control Register will not be used.
8	CH0DlyEn	RW 0x0	Channel 0 Delay Control Enable 0 = Disable: Disable the use of <CH0_RX_DLY> and <CH0_TX_DLY> field in the Channel 0 Delay Control Register. The value in <CH0_TX_TSLOT> and <CH0_RX_TSLOT> field in the Channel Time Slot Control Register will be used instead for receive and transmit for channel 0 data. 1 = Enable: Enable the use of Channel 0 delay control <CH0_RX_DLY> and <CH0_TX_DLY> field in the Channel 0 Delay Control Register. The value in <CH0_TX_TSLOT> and <CH0_RX_TSLOT> field in the Channel Time Slot Control Register will not be used.
7	Reserved	RW 0x0	Reserved Always write 0 to this bit.
6	PcmSampleSize	RW 0x0	PCM Sample Size 0 = 1 Byte: Each sample is 1 byte, and occupies one time slot. 1 = 16-bit: In this mode (Linear mode), each sample is 16-bits, and occupies two time slots.
5	LongFs	RW 0x0	In Frame Sync master mode, the TDM generates frame sync (FS). This bit defines the type of frame sync generated.
4	InvFs	RW 0x0	Inverted Frame Sync When this bit is set, FS is interpreted as Active Low. Only valid when using Short FS.
3	FsPol	RW 0x0	Frame sync Reference Polarity is referred as the clock edge where the FS is sampled asserted or de-asserted by the external CODEC. The TDM drives the FS at the opposite edge from what is sampled by CODEC. Frame sync Reference Polarity: This bit and <DataPol> should be changed together. This bit cannot be changed without changing the <DataPol> . .

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 326: PCM Control Register (Continued)
Offset: 0x000B0000

Bit	Field	Type/InitVal	Description
2	DataPol	RW 0x0	Data Reference Polarity Data Reference Polarity is referred as the clock edge where the data is sampled asserted or de-asserted by the external CODEC. The TDM drives the data at the opposite edge from what is sampled by the CODEC. Data Reference Polarity: This bit can be changed without changing <FsPol> .
1	MasterFsn	RW 0x1	TDM is Master/Slave of Frame Sync. 1 = Use external FS. The TDM will get FS from some external source. 0 = Use internally generated FS.
0	MstrPclkn	RW 0x1	TDM is Master/Slave of PCM clock 1 = Use external PCM clock. The TDM will get PCM clock from some external source. 0 = Use internally generated PCM clock. See TDM PCM Clock Rate Divisor Register for details of PCM clock supported by TDM.

Table 327: Channel Time Slot Control Register
Offset: 0x000B0004

Bit	Field	Type/InitVal	Description
NOTE: TDM supports a maximum 128 time slot (at 8.192 MHz PCM clock). Transmit and receive of Channel 0 and Channel 1 can be assigned to any one of the 128 time slot. The time slot assignment for Channel 0 transmit/receive and Channel 1 transmit/receive must be mutually exclusive. If these assignments overlap, contention will occur on the PCM bus. Channel 0 Tx/Rx and Channel 1 Tx/Rx can be assigned a time slot or a delay value (from assertion of frame sync). The delay is programmed via the Channel x Delay Control Register / and enabled via the PCM Control Register . If the Channel x Delay Control Register value is enabled in the PCM Control Register, then the Channel time slot control register value is not used for the Tx and Rx for that channel. The TDM Tx time slot is CODEC Rx time slot and vice versa.			
31:24	CH1TxTSlot	RW 0x02	Transmit (TX from TDM to CODEC) time slot for channel 1 See the "Number of Time Slots Register" for available time slots at a particular PCM clock frequency. If the time slot value programmed is more than the available time slots, the TDM will not transmit data. If <PCM_SAMPLE_SIZE> field in the PCM Control Register is set to 1, two time slots are used for TX data.
23:16	CH1RxTSlot	RW 0x02	Receive (RX from CODEC to TDM) time slot for channel 1 See the 'Number of Time Slots Register" for available time slots at a particular PCM clock frequency. If the time slot value programmed is more than the available time slots, the TDM will not receive data. If <PCM_SAMPLE_SIZE> is set to 1, two time slots are used for RX data.
15:8	CH0TxTSlot	RW 0x00	Transmit (TX from TDM to CODEC) time slot for channel 0 See table 16 for available time slots at particular PCM clock frequency. If the time slot value programmed is more than the available time slots, the TDM will not transmit data. If <PCM_SAMPLE_SIZE> is set to 1, two time slots are used for TX data.
7:0	CH0RxTSlot	RW 0x00	Receive (RX from CODEC to TDM) time slot for channel 0 See "Number of Time Slots Register" for available time slots at a particular PCM clock frequency. If the time slot value programmed is more than the available time slots, the TDM will not receive data. If <PCM_SAMPLE_SIZE> is set to 1, two time slots are used for RX data.

Copyright © 2015 Marvell

Table 328: Channel 0 Delay Control Register
Offset: 0x000B0008

Bit	Field	Type/InitVal	Description
NOTE: The Channel0/1 receive and transmit delay (from frame sync assertion) can be controlled by this register. The delay is defined in number of PCLK from the assertion of FS. The value in this register overrides the value in CH0TxTSlot/CH1TxTSlot and CH0RxTSlot/CH1RxTSlot field in the Channel Time Slot Control Register if <CH0DlyEn> /<CH1DlyEn> field in the PCM Control Register is set to 1.			
31:26	Reserved	RSVD 0x0	Reserved
25:16	CHn0TxDly	RW 0x00	Number of PCLK to wait before starting to transmit data for Channel 0 from the assertion of FS. After the start of transmission, 8-bit or 16-bit data will be transmitted depending on the <PCM_SAMPLE_SIZE> field in the PCM Control Register. This bit also controls the time when the first PCM data transmit is done in Wideband mode in CH0.
15:10	Reserved	RSVD 0x0	Reserved
9:0	CH0RxDly	RW 0x00	Number of PCLK to wait before starting to receive data for Channel 0 the assertion of FS. After the start of receive, 8-bit or 16-bit data will be received depending on the <PCM_SAMPLE_SIZE>. This bit also controls the time when the first PCM data receive occurs in Wideband mode in CH0.

Table 329: Channel 1 Delay Control Register
Offset: 0x000B000C

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25:16	Ch1TxDly	RW 0x10	Number of PCLK to wait before starting to transmit data for Channel 1 from the assertion of FS. After the start of transmission, 8-bit or 16-bit data will be transmitted depending on the <PCM_SAMPLE_SIZE> field in the PCM Control Register. This bit also controls the time when the first PCM data transmit is done in Wideband mode in CH1.
15:10	Reserved	RSVD 0x0	Reserved
9:0	CH1RxDly	RW 0x10	Number of PCLK to wait before starting to receive data for Channel 1 from assertion of FS. After the start of receive, 8-bit or 16-bit data will be received depending on the <PCM_SAMPLE_SIZE>. This bit also controls the time when the first PCM data receive occurs in Wideband mode in CH1.

Copyright © 2015 Marvell

Table 330: Channel n Enable and Disable Register (n=0–1)
Offset: Channel0: 0x000B0010, Channel1: 0x000B0020
Offset Formula: 0x000B0010+n*0x10: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
NOTE: After the Channel 0/1 RX/TX bits are enabled the design starts to receive or transmit data in the programmed time slot (or delay) after the next FS is detected. Before programming the different enable bits in the Channel0 Delay Control Register or Channel1 Delay Control Register , TDM channel 0/1 should be initialized. See details of initialization in PCM data flow for TX and RX. The software can disable the transmit operation of any channel any time after enabling it. The disable only takes affect after all the PCM data in all the allocated TX buffers in memory is sent out to the CODEC. In case of receive, if the channel is disabled, the design stops to receive data from the PCM interface and starts padding the remaining PCM samples from the Dummy Data for Dummy RX Write Register. The transmit enable or receive enable is automatically disabled if an underflow (in case of transmit), or an overflow (in case of receive) situation is detected by the hardware. Refer to TDM underflow/overflow handling section for or more detail. If the channel is not enabled, the TDM transmit 0/1 in allocated a TX time slot and does not receive any data in the RX time slot.			
31:9	Reserved	RSVD 0x0	Reserved
8	CHnTxEn	RW 0x0	0 = Disable: Disable channel 0/1 transmit 1 = Enable: Enable channel 0/1 transmit
7:1	Reserved	RSVD 0x0	Reserved
0	CHnRxEn	RW 0x0	0 = Disable: Disable channel 0/1 receive 1 = Enable: Enable channel 0/1 receive

Table 331: Channel n Buffer Ownership Register (n=0–1)
Offset: Channel0: 0x000B0014, Channel1: 0x000B0024
Offset Formula: 0x000B0014+n*0x10: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
NOTE: This register defines the ownership of the Transmit Data Start Address for Channel 0/1 Register and Receive Data Start Address for Channel 0/1 Register register. By default, the software is the owner and should set the respective bit to 1 once a valid address has been programmed in the Transmit Data Start Address for Channel 0/1 Register or Receive Data Start Address for Channel 0/1 Register. The hardware will reset the respective bit to 0 once it makes a copy of the address for the current buffer. Software should read this bit before programming the Transmit Data Start Address for Channel 0/1 Register or Receive Data Start Address for Channel 0/1 Register. Using this mechanism, the software can implement a ping-pong buffer scheme in the memory for Channel 0/1 receive and transmit data buffers.			
31:9	Reserved	RSVD 0x0	Reserved
8	TX_DMA_ST_ADD R_OWN_CHn	RW 0x0	This register defines the ownership of the Transmit Data Start Address for Channel x Register .
7:1	Reserved	RSVD 0x0	Reserved
0	RX_DMA_ST_ADD R_OWN_ CHn	RW 0x0	This register defines the ownership of the Receive Data Start Address for Channel x Register .

Table 332: Channel 0 Transmit Data Start Address Register
Offset: 0x000B0018

Bit	Field	Type/InitVal	Description
31:0	TX_DMA_ST_ADDR_CH0	RW 0xC0000000	This register contains the start address of the allocated buffer for Channel0 (CH0) transmit data. Software should check the <TX_DMA_ST_ADDR_OWN_CHx> field in the Buffer Ownership for Channel x Register before programming this register. **NOTE:** Start address must be 32-byte aligned.

Table 333: Channel 0 Receive Data Start Address Register
Offset: 0x000B001C

Bit	Field	Type/InitVal	Description
31:0	RX_DMA_ST_ADDR_CH0	RW 0xC0001000	This register contains the start address of the allocated buffer for CH0 receive data. Software should check the <RX_DMA_ST_ADDR_OWN_ CHx> field in the Buffer Ownership for Channel x Register before programming this register. **NOTE:** Start address must be 32-byte aligned.

Table 334: Channel 1 Transmit Data Start Address Register
Offset: 0x000B0028

Bit	Field	Type/InitVal	Description
31:0	TX_DMA_ST_ADDR_CH1	RW 0xC0002000	This register contains the start address of the allocated buffer for Channel1 (CH1) transmit data. Software should check the <TX_DMA_ST_ADDR_OWN_CHx> field in the Buffer Ownership for Channel x Register before programming this register. **NOTE:** Start address must be 32-byte aligned.

Table 335: Channel 1 Receive Data Start Address Register
Offset: 0x000B002C

Bit	Field	Type/InitVal	Description
31:0	RX_DMA_ST_ADDR_CH1	RW 0xC0003000	This register contains the start address of the allocated buffer for CH1 receive data. Software should check the <RX_DMA_ST_ADDR_OWN_ CHx> field in the Buffer Ownership for Channel x Register before programming this register. **NOTE:** Start address must be 32-byte aligned.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 336: Channel n Total Sample Count Register (n=0–1)
Offset: Channel0: 0x000B0030, Channel1: 0x000B0034
Offset Formula: 0x000B0030+n*0x4: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
NOTE: This register sets the total number of sample (8-bit or 16-bit each) in the allocated buffer to be transmitted or received by Channel 0/1 (if enabled) and the number of samples after which the TDM will generate an interrupt to the CPU indicating the hardware is close to the end of total sample count value to make the next buffer ready. The sample count is programmed before enabling the channel and cannot be changed until the channel is disabled. The value is copied internally after the channel is enabled and decremented by one with every FS detection (as the PCM data TX and RX starts). The hardware uses the interrupt sample count value (<CHx_INT_SMPL_CNT> field in the Total Sample Count Channel x Register) as a watermark for the availability of new buffers to continue to move data across the PCM interface.			
31:16	Reserved	RSVD 0x0	Reserved
15:8	CHn_INT_SMPL_CNT	RW 0x40	Interrupt Sample count can be defined as the number of samples transferred to CODEC or received from CODEC after which the hardware will notify the software to create the next buffer. This register should be programmed with a value which is less than the value in the <CHx_TOTAL_ SMPL_CNT> field in the Total Sample Count Channel x Register . The value should take into consideration the GBUS latencies and software latencies to make the next buffer available. See the detailed explanation in the "Tx Data Flow" and "Rx Data Flow".
7:0	CHn_TOTAL_SMPL_CNT	RW 0x50	Total Sample count. This register needs to be programmed if the buffer size is different than 80 samples. This can be programmed for up to 30 ms sample width (240 samples). **NOTE:** The buffer size depends upon whether each PCM sample is 1 byte or 2 bytes long. As an example, if this bit is set to 80 and each PCM sample is 1 byte long, the buffer size will be 80 bytes while if each PCM sample is 2 byte long it will be 160 bytes.

Table 337: Number of Time Slots Register
Offset: 0x000B0038

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	NO_OF_TS	RW 0x80	This register contains the number of time slots (bytes) in a frame. This value is used to generate Frame sync signals in Frame sync master mode of TDM. Default is 128 considering the bit clock is 8.192 MHz. In Frame sync slave mode, this register does not mean anything. Number of time slots = PCLK (MHz) **NOTE:** Any other values are reserved 4 = 256 kHz 8 = 512 kHz 16 = 1.024 MHz 32 = 2.048 MHz 64 = 4.096 MHz 128 = 8.192 MHz

Copyright © 2015 Marvell

Table 338: TDM PCM Clock Rate Divider Register
Offset: 0x000B003C

Bit	Field	Type/InitVal	Description
NOTE: When the TDM is configured to be the clock master, this register sets the PCM clock baud rate. The BBU_PCM_CLK is divided according to the information in this register and gives the frequency of PCM Clock. The divided ratios support only one hot encoded values (implemented in BBU).			
31:8	Reserved	RSVD 0x0	Reserved
7:0	PCM_CLK_DIV	RW 0x20	PCM clock Rate Divider PCM_CLK_DIV = PCM CLOCK Frequency **NOTE:** Any other values are reserved 1 = 128 Khz 2 = 256 Khz 4 = 512 Khz 8 = 1024 Khz 16 = 2048 Khz 32 = 4096 Khz 64 = 8192 Khz

Table 339: Interrupt Event Mask Register
Offset: 0x000B0040

Bit	Field	Type/InitVal	Description
31:18	Reserved	RW 0x0	Reserved
17:0	IEMR	RW 0x3FFFF	Interrupt event mask register

Table 340: Interrupt Status Mask Register
Offset: 0x000B0048

Bit	Field	Type/InitVal	Description
NOTE: This is a dummy register to facilitate the Interrupt latching in the Interrupt Status Register.			
31:18	Reserved	RW 0x0	Reserved
17:0	ISM	RW 0x3FFFF	Interrupt mask register 0 = The interrupt is masked: The interrupt is masked. To check the interrupt, poll the Interrupt Status Register (TDM_INT_STATUS). 1 = No interrupt is masked: The corresponding bit in the Interrupt Status Register causes an interrupt to CPU. By default, no interrupt is masked.

Copyright © 2015 Marvell

Table 341: Interrupt Reset Selection Register
Offset: 0x000B004C

Bit	Field	Type/InitVal	Description
31:18	Reserved	RW 0x0	Reserved
17:0	IRS	RW 0x0	Interrupt reset select register configures the way the ISR bit will clear after being set.

Table 342: Interrupt Status Register
Offset: 0x000B0050

Bit	Field	Type/InitVal	Description
31:18	Reserved	RSVD 0x0	Reserved
17	CODEC_INT	RW 0x0	CODEC interrupt
16	DMA_ ABORTED_INT	RW 0x0	TDM DMA has aborted. This is caused by illegal memory space access by TDM.
15	TXFIFO1_ EMPTY_INT	RW 0x0	TXFIFO empty for Channel 1 interrupt
14	RXFIFO1_ FULL_INT	RW 0x0	RXFIFO full for Channel 1 interrupt
13	TXFIFO0_ EMPTY_INT	RW 0x0	TXFIFO empty for Channel 0 interrupt
12	RXFIFO0_ FULL_INT	RW 0x0	RXFIFO full for Channel 0 interrupt
11	CH1_TX_IDLE	RW 0x0	Channel 1 transmit has gone from active state to idle. This occurs when the total sample count Channel 1 is reached, no more buffers are pending and channel has been closed by programming the <CHxTxEn> field in the Channel x Enable and Disable Register. This interrupt will not be generated when the channel 1 get closed by itself because of underflow.
10	CH1_RX_IDLE	RW 0x0	Channel 1 receive has gone from active state to idle. This occurs when the total sample count Channel 1 is reached and channel has been closed by programming the <CHxRxEn> field in the Channel x Enable and Disable Register. This interrupt will not be generated when the Channel 1 get closed by itself because of overflow.
9	CH0_TX_IDLE	RW 0x0	Channel 0 transmit has gone from active state to idle This occurs when the total sample count Channel 0 is reached, no more buffers are pending and channel has been closed by programming the <CHxTxEn> field in the Channel x Enable and Disable Register . This interrupt will not be generated when the Channel 0 get closed by itself because of underflow.

Copyright © 2015 Marvell

Table 342: Interrupt Status Register (Continued)
Offset: 0x000B0050

Bit	Field	Type/InitVal	Description
8	CH0_RX_IDLE	RW 0x0	Channel 0 receive has gone from active state to idle This occurs when the total sample count Channel 0 is reached and channel has been closed by programming the <CHxRxEn> field in the Channel x Enable and Disable Register. This interrupt will not be generated when the Channel 0 get closed by itself because of overflow.
7	SCOCH1_TX_ INT	RW 0x0	Sample count over as per <CHx_INT_SMPL_CNT> field in the Total Sample Count Channel x Register value for Channel 1
6	SCOCH1_RX_ INT	RW 0x0	Sample count over as per <CHx_INT_SMPL_CNT> field in the Total Sample Count Channel x Register value for Channel 1
5	SCOCH0_TX_ INT	RW 0x0	Sample count over as per <CHx_INT_SMPL_CNT> field in the Total Sample Count Channel x Register value for Channel 0
4	SCOCH0_RX_ INT	RW 0x0	Sample count over as per <CHx_INT_SMPL_CNT> field in the Total Sample Count Channel x Register value for Channel 0
3	UFLOW_CH1_ INT	RW 0x0	Transmit Underflow in CH1
2	OFLOW_CH1_ INT	RW 0x0	Receive Overflow in CH1
1	UFLOW_CH0_ INT	RW 0x0	Transmit Underflow in CH0
0	OFLOW_CH0_ INT	RW 0x0	Receive Overflow in CH0

Table 343: Dummy Data for Dummy RX Write Register
Offset: 0x000B0054

Bit	Field	Type/InitVal	Description
31:0	DUMMY_DATA	RW 0x00	Contains the value of the padding that will be used by the TDM when RX is switched off by programming the <CHxRxEn> field in the Channel x Enable and Disable Register. The TDM will stop receiving data from PCM interface and fill the remaining allocated buffer with this dummy data.

Table 344: Miscellaneous Control Register
Offset: 0x000B0058

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	CODEC_RST	RW 0x0	Controls the CODEC reset signal which is output form TDM to CODEC. The CODEC reset is active low reset which should be asserted for sufficient time so that CODEC can sample PCM clock and Frame Sync to lock it's internal PLL (See individual CODEC specs for more details).

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 345: Channel n Transmit Data Current Address Register (n=0–1)
Offset: Channel0: 0x000B0060, Channel1: 0x000B0068
Offset Formula: 0x000B0060+n*0x8: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
31:0	TX_DMA_CUR_ADDR_CHn	RO 0x00	Contains the current address of the allocated buffer for CH0/1 transmit DMA. Used for debug purposes.

Table 346: Channel n Receive Data Current Address Register (n=0–1)
Offset: Channel0: 0x000B0064, Channel1: 0x000B006C
Offset Formula: 0x000B0064+n*0x8: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
31:0	RX_DMA_CUR_ADDR_CHn	RO 0x0	Contains the current address of the allocated buffer for CH0/1 receive DMA. Used for debug purposes.

Table 347: Current Time Slot Register
Offset: 0x000B0070

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	CUR_TS	RO 0x0	Current Time Slot Identifies the time slot that is currently active. This can be used by FW as a check for precise time for changing control register of TDM while it is in operation.

Table 348: TDM Revision Register
Offset: 0x000B0074

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	TDM_IP_REV	RO 0x0201	TDM IP revision

Table 349: TDM Channel n Debug Register (n=0–1)
Offset: Channel0: 0x000B0078, Channel1: 0x000B007C
Offset Formula: 0x000B0078+n*0x4: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
31:24	CUR_SAMPLE_CNT_CHn_TX	RO 0x0	Current sample count for CH0/1 TX
23	REQ_CHn_TX	RO 0x0	CH0/1 TX request to TDM-DMA

Copyright © 2015 Marvell

Table 349: TDM Channel n Debug Register (n=0–1) (Continued)
Offset: Channel0: 0x000B0078, Channel1: 0x000B007C
Offset Formula: 0x000B0078+n*0x4: where n (0-1) represents Channel

Bit	Field	Type/InitVal	Description
22	GNT_CHn_TX	RO 0x0	CH0/1 TX grant
21:19	PCM_TX_IF_ STATE_CHn_TX	RO 0x0	TX PCM state machine state
18:16	TX_RD_CTRL_ STATE_CHn_TX	RO 0x0	Tx read control state machine
15:8	CUR_SAMPLE_CN T_CHn_RX	RO 0x0	Current sample count for CH0/1 RX
7	OVERFLOW_ PENDING_CHn_RX	RO 0x0	Overflow flag in CH0/1
6	REQ_CHn_RX	RO 0x0	CH0/1 RX request to GbDMA
5	GNT_CHn_RX	RO 0x0	CH0/1 RX grant
4:2	PCM_RX_IF_ STATE_CHn_RX	RO 0x0	RX PCM state machine state
1:0	RX_WR_CTRL_STA TE_CHn_RX	RO 0x0	RX read control state machine

Table 350: TDM DMA Abort Register 1
Offset: 0x000B0080

Bit	Field	Type/InitVal	Description
31:0	TDM_DMA_ABORT _ADDRESS	RO 0x0	In case of illegal GBUS address space access by TDM DMA, the DMA will abort and the address which was accessed will be latched in this register. Also, the TDM DMA Abort Register 2 will contain other DMA parameters. For debug purposes only.

Table 351: TDM DMA Abort Register 2
Offset: 0x000B0084

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28	DMA_CYCLE_TYP E_ ABORTED	RO 0x0	Read or write cycle DMA
27:16	DMA_ID_ ABORTED	RO 0x0	DMA ID when it was aborted.
15:0	DMA_ TRANSFER_ LENGTH_ ABORTED	RO 0x0	DMA transfer length when it was aborted.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 352: TDM Channel 0 Wideband Delay Control Register
Offset: 0x000B0088

Bit	Field	Type/InitVal	Description
NOTE: Channel 0/1 receive and transmit delay (from frame sync assertion) is controlled by the TDM Channel 0/1 Wideband Delay Control Register for second PCM sample in Wideband mode. The delay is defined in the number of PCLK from the assertion of FS. For Channel 0 <CH0DlyEn> must be set to 1 in the PCM Control Register. For Channel 1 <CH1DlyEn> must be set to 1 in the PCM Control Register			
31:26	Reserved	RSVD 0x0	Reserved
25:16	CH0_WBAND_TX_DLY	RW 0x40	Number of PCLK cycles to wait before starting to transmit a second PCM sample for Wideband mode for Channel 0 from the assertion of FS. After the start of transmission, 8-bit or 16-bit data will be transmitted depending on the setting of the <PcmSampleSize> field in the PCM Control Register.
15:10	Reserved	RSVD 0x0	Reserved
9:0	CH0_WBAND_RX_DLY	RW 0x40	Number of PCLK cycles to wait before starting to receive a second PCM sample for Wideband mode for Channel 0 from the assertion of Frame Sync (FS). After the start of receive, 8-bit or 16-bit data will be received depending on the setting of the <PcmSampleSize> field in the PCM Control Register.

Table 353: TDM Channel 1 Wideband Delay Control Register
Offset: 0x000B008C

Bit	Field	Type/InitVal	Description
NOTE: Channel 0/1 receive and transmit delay (from frame sync assertion) is controlled by the TDM Channel 0/1 Wideband Delay Control Register for second PCM sample in Wideband mode. The delay is defined in the number of PCLK from the assertion of FS. For Channel 0 <CH0DlyEn> must be set to 1 in the PCM Control Register. For Channel 1 <CH1DlyEn> must be set to 1 in the PCM Control Register			
31:26	Reserved	RSVD 0x0	Reserved
25:16	CH1_WBAND_TX_DLY	RW 0x40	Number of PCLK cycles to wait before starting to transmit a second PCM sample for Wideband mode for Channel 1 from the assertion of FS. After the start of transmission, 8-bit or 16-bit data will be transmitted depending on the setting of the <PcmSampleSize> field in the PCM Control Register.
15:10	Reserved	RSVD 0x0	Reserved
9:0	CH1_WBAND_RX_DLY	RW 0x40	Number of PCLK cycles to wait before starting to receive a second PCM sample for Wideband mode for Channel 1 from the assertion of Frame Sync (FS). After the start of receive, 8-bit or 16-bit data will be received depending on the setting of the <PcmSampleSize> field in the PCM Control Register.

Copyright © 2015 Marvell

Copyright © 2015 Marvell

A.7 PCI Express 2.0 Port 0 Registers

The following table provides a summarized list of all registers that belong to the PCI Express 2.0 Port 0, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 354: Summary Map Table for the PCI Express 2.0 Port 0 Registers

Register Name	Offset	Table and Page
PCI Express Address Window Control (Port 0)		
PCI Express Window0 Control Register	0x00081820	Table 355, p. 574
PCI Express Window0 Base Register	0x00081824	Table 356, p. 574
PCI Express Window0 Remap Register	0x0008182C	Table 357, p. 575
PCI Express Window1 Control Register	0x00081830	Table 358, p. 575
PCI Express Window1 Base Register	0x00081834	Table 359, p. 575
PCI Express Window1 Remap Register	0x0008183C	Table 360, p. 576
PCI Express Window2 Control Register	0x00081840	Table 361, p. 576
PCI Express Window2 Base Register	0x00081844	Table 362, p. 577
PCI Express Window2 Remap Register	0x0008184C	Table 363, p. 577
PCI Express Window3 Control Register	0x00081850	Table 364, p. 577
PCI Express Window3 Base Register	0x00081854	Table 365, p. 578
PCI Express Window3 Remap Register	0x0008185C	Table 366, p. 578
PCI Express Window4 Control Register	0x00081860	Table 367, p. 578
PCI Express Window4 Base Register	0x00081864	Table 368, p. 579
PCI Express Window4 Remap Register	0x0008186C	Table 369, p. 579
PCI Express Window4 Remap (High) Register	0x00081870	Table 370, p. 579
PCI Express Window5 Control Register	0x00081880	Table 371, p. 579
PCI Express Window5 Base Register	0x00081884	Table 372, p. 580
PCI Express Window5 Remap Register	0x0008188C	Table 373, p. 580
PCI Express Window5 Remap (High) Register	0x00081890	Table 374, p. 581
PCI Express Default Window Control Register	0x000818B0	Table 375, p. 581
PCI Express Expansion ROM Window Control Register	0x000818C0	Table 376, p. 581
PCI Express Expansion ROM Window Remap Register	0x000818C4	Table 377, p. 582
PCI Express Window<n+6> Control Register (n=0–9)	x10: 0x00081C00, x11: 0x00081C10...x19: 0x00081C90	Table 378, p. 582
PCI Express Window<n+6> Base Register (n=0–9)	x10: 0x00081C04, x11: 0x00081C14...x19: 0x00081C94	Table 379, p. 583
PCI Express Window<n+6> Remap Register (n=0–9)	x10: 0x00081C08, x11: 0x00081C18...x19: 0x00081C98	Table 380, p. 583
PCI Express Window<n+6> Remap (High) Register (n=0–9)	x10: 0x00081C0C, x11: 0x00081C1C...x19: 0x00081C9C	Table 381, p. 583
PCI Express BAR Control (Port 0)		
PCI Express BAR1 Control Register	0x00081804	Table 382, p. 583
PCI Express BAR2 Control Register	0x00081808	Table 383, p. 584

Table 354: Summary Map Table for the PCI Express 2.0 Port 0 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Expansion ROM BAR Control Register	0x0008180C	Table 384, p. 584
PCI Express Configuration Cycles Generation (Port 0)		
PCI Express Configuration Address Register	0x000818F8	Table 385, p. 585
PCI Express Configuration Data Register	0x000818FC	Table 386, p. 585
PCI Express Configuration Header (Port 0)		
PCI Express Device and Vendor ID Register	0x00080000	Table 387, p. 585
PCI Express Command and Status Register	0x00080004	Table 388, p. 586
PCI Express Class Code and Revision ID Register	0x00080008	Table 389, p. 588
PCI Express BIST Header Type and Cache Line Size Register	0x0008000C	Table 390, p. 588
PCI Express BAR0 Internal Register	0x00080010	Table 391, p. 588
PCI Express BAR0 Internal (High) Register	0x00080014	Table 392, p. 589
PCI Express BAR1 Register	0x00080018	Table 393, p. 589
PCI Express BAR1 (High) Register	0x0008001C	Table 394, p. 590
PCI Express BAR2 Register	0x00080020	Table 395, p. 590
PCI Express BAR2 (High) Register	0x00080024	Table 396, p. 590
PCI Express Subsystem Device and Vendor ID Register	0x0008002C	Table 397, p. 590
PCI Express Expansion ROM BAR Register	0x00080030	Table 398, p. 591
PCI Express Capability List Pointer Register	0x00080034	Table 399, p. 591
PCI Express Interrupt Pin and Line Register	0x0008003C	Table 400, p. 591
PCI Express Power Management Capability Header Register	0x00080040	Table 401, p. 592
PCI Express Power Management Control and Status Register	0x00080044	Table 402, p. 592
PCI Express MSI Message Control Register	0x00080050	Table 403, p. 594
PCI Express MSI Message Address Register	0x00080054	Table 404, p. 594
PCI Express MSI Message Address (High) Register	0x00080058	Table 405, p. 594
PCI Express MSI Message Data Register	0x0008005C	Table 406, p. 595
PCI Express Capability Register	0x00080060	Table 407, p. 595
PCI Express Device Capabilities Register	0x00080064	Table 408, p. 595
PCI Express Device Control Status Register	0x00080068	Table 409, p. 597
PCI Express Link Capabilities Register	0x0008006C	Table 410, p. 599
PCI Express Link Control Status Register	0x00080070	Table 411, p. 601
PCI Express Device Capabilities 2 Register	0x00080084	Table 412, p. 603
PCI Express Device Control Status 2 Register	0x00080088	Table 413, p. 603
PCI Express Link Control Status 2 Register	0x00080090	Table 414, p. 604
PCI Express Advanced Error Report Header Register	0x00080100	Table 415, p. 606
PCI Express Uncorrectable Error Status Register	0x00080104	Table 416, p. 607
PCI Express Uncorrectable Error Mask Register	0x00080108	Table 417, p. 607
PCI Express Uncorrectable Error Severity Register	0x0008010C	Table 418, p. 608
PCI Express Correctable Error Status Register	0x00080110	Table 419, p. 609
PCI Express Correctable Error Mask Register	0x00080114	Table 420, p. 610

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 354: Summary Map Table for the PCI Express 2.0 Port 0 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Advanced Error Capability and Control Register	0x00080118	Table 421, p. 611
PCI Express Header Log First DWORD Register	0x0008011C	Table 422, p. 611
PCI Express Header Log Second DWORD Register	0x00080120	Table 423, p. 611
PCI Express Header Log Third DWORD Register	0x00080124	Table 424, p. 612
PCI Express Header Log Fourth DWORD Register	0x00080128	Table 425, p. 612
PCI Express Root Error Command Register	0x0008012C	Table 426, p. 612
PCI Express Root Error Status Register	0x00080130	Table 427, p. 612
PCI Express Error Source Identification Register	0x00080134	Table 428, p. 613
PCI Express Control and Status (Port 0)		
PCI Express Control Register	0x00081A00	Table 429, p. 613
PCI Express Status Register	0x00081A04	Table 430, p. 615
PCI Express Spare Register	0x00081A08	Table 431, p. 616
PCI Express Root Complex Set Slot Power Limit (SSPL) Register	0x00081A0C	Table 432, p. 616
PCI Express Completion Timeout Register	0x00081A10	Table 433, p. 617
PCI Express Root Complex Power Management Event (PME) Register	0x00081A14	Table 434, p. 617
PCI Express Power Management Register	0x00081A18	Table 435, p. 618
PCI Express Flow Control Register	0x00081A20	Table 436, p. 619
PCI Express Replay Timeout (TO) Timer Register	0x00081A24	Table 437, p. 619
PCI Express Ack Latency Timer Register	0x00081A28	Table 438, p. 619
PCI Express Secondary Bus Register	0x00081A2C	Table 439, p. 620
PCI Express Debug (Port 0)		
PCI Express Debug Control Register	0x00081A60	Table 440, p. 620
PCI Express Debug MAC Status 1 Register	0x00081A84	Table 441, p. 621
PCI Express TL Control Register	0x00081AB0	Table 442, p. 621
PCI Express Interrupt (Port 0)		
PCI Express Interrupt Cause Register	0x00081900	Table 443, p. 622
PCI Express Interrupt Mask Register	0x00081910	Table 444, p. 625
PCI Express Mbus Control (Port 0)		
PCI Express Mbus Adapter Control Register	0x000818D0	Table 445, p. 625
PCI Express Mbus Arbiter Control Register (Low)	0x000818E0	Table 446, p. 626
PCI Express Mbus Arbiter Control Register (High)	0x000818E4	Table 447, p. 627
PCI Express Mbus Arbiter Timeout Register	0x000818E8	Table 448, p. 627

Copyright © 2015 Marvell

A.7.1 PCI Express Address Window Control (Port 0)

Table 355: PCI Express Window0 Control Register
Offset: 0x00081820

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by a sequence of 0s. The number of 1s specifies the size of the window in 64-KByte granularity. (A value of 0x0FFF specifies 256 MByte).
15:8	Attr	RW 0x0E	Target specific attributes depending on the target interface. See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See the "PCI Express Address Decoding" section.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping to BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Table 356: PCI Express Window0 Base Register
Offset: 0x00081824

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0000	Base Address Used with the <Size> field to set the address window size and location. Corresponds to transaction address [31:16].
15:0	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information

Table 357: PCI Express Window0 Remap Register
Offset: 0x0008182C

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address Used with the <Size> field to specifies address bits[31:0] to be driven to the target interface.
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 358: PCI Express Window1 Control Register
Offset: 0x00081830

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size (A value of 0x0FFF specifies 256 MByte).
15:8	Attr	RW 0x0D	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable. 0 = Disable 1 = Enable

Table 359: PCI Express Window1 Base Register
Offset: 0x00081834

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x1000	Base Address

Table 359: PCI Express Window1 Base Register (Continued)
Offset: 0x00081834

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 360: PCI Express Window1 Remap Register
Offset: 0x0008183C

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 361: PCI Express Window2 Control Register
Offset: 0x00081840

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size
15:8	Attr	RW 0x0B	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 362: PCI Express Window2 Base Register
Offset: 0x00081844

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x2000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 363: PCI Express Window2 Remap Register
Offset: 0x0008184C

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 364: PCI Express Window3 Control Register
Offset: 0x00081850

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size
15:8	Attr	RW 0x07	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = 128B: Write split on 128-Byte address boundary. 1 = 32B: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable. Window is disabled by default. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 365: PCI Express Window3 Base Register
Offset: 0x00081854

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x3000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 366: PCI Express Window3 Remap Register
Offset: 0x0008185C

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 367: PCI Express Window4 Control Register
Offset: 0x00081860

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	Window Size (A value of 0x07FF specifies 128 MByte).
15:8	Attr	RW 0x3B	Target specific attributes depending on the target interface.
7:4	Target	RW 0x1	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = 128B: Write split on 128-Byte address boundary. 1 = 32B: Write split on 32-Byte address boundary.
1	BarMap	RW 0x1	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 368: PCI Express Window4 Base Register
Offset: 0x00081864

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xF000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 369: PCI Express Window4 Remap Register
Offset: 0x0008186C

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 370: PCI Express Window4 Remap (High) Register
Offset: 0x00081870

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap High Address Specifies address bits[63:32] to be driven to the target interface. Only relevant for target interfaces that support a more than 4-GByte address space. When using target interface that do not support a more than 4-GByte address space, this register must be cleared.

Table 371: PCI Express Window5 Control Register
Offset: 0x00081880

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	Window Size (A value of 0x07FF specifies 128 MByte).
15:8	Attr	RW 0x2F	Target specific attributes depending on the target interface.
7:4	Target	RW 0x1	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 371: PCI Express Window5 Control Register (Continued)
Offset: 0x00081880

Bit	Field	Type/InitVal	Description
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = 128B: Write split on 128-Byte address boundary. 1 = 32B: Write split on 32-Byte address boundary.
1	BarMap	RW 0x1	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable 0 = Disable 1 = Enable

Table 372: PCI Express Window5 Base Register
Offset: 0x00081884

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xF800	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 373: PCI Express Window5 Remap Register
Offset: 0x0008188C

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Document Classification: Proprietary Information

Table 374: PCI Express Window5 Remap (High) Register
Offset: 0x00081890

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap High Address Specifies address bits[63:32] to be driven to the target interface. Only relevant for target interfaces that support a more than 4-GByte address space. When using target interface that do not support a more than 4-GByte address space, this register must be cleared.

Table 375: PCI Express Default Window Control Register
Offset: 0x000818B0

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	Attr	RW 0x0	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1:0	Reserved	RSVD 0x0	Reserved

Table 376: PCI Express Expansion ROM Window Control Register
Offset: 0x000818C0

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	Attr	RW 0x2F	Target specific attributes depending on the target interface. Default corresponds to DEV_BOOTCSn.
7:4	Target	RW 0x1	Specifies the unit ID (target interface) associated with this window. Default represents the Device Bus controller.
3:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 377: PCI Express Expansion ROM Window Remap Register
Offset: 0x000818C4

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 378: PCI Express Window<n+6> Control Register (n=0–9)
Offset: x10: 0x00081C00, x11: 0x00081C10...x19: 0x00081C90
Offset Formula: 0x00081C00+0x10*n: where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by a sequence of 0s. The number of 1s specifies the size of the window in 64-KByte granularity. (A value of 0x0FFF specifies 256 MByte).
15:8	Attr	RW 0x0E	Target specific attributes depending on the target interface. See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See the "PCI Express Address Decoding" section.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control When forwarded to the internal interface, controls the address boundary on which slave write transactions are split. Only relevant for transactions that hit the respective address window. **NOTE:** When working with cache coherency, this bit must be set to 1. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping to BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x0	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Copyright © 2015 Marvell

Table 379: PCI Express Window<n+6> Base Register (n=0–9)
Offset: x10: 0x00081C04, x11: 0x00081C14...x19: 0x00081C94
Offset Formula: 0x00081C04+n*0x10: where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0000	Base Address Used with the <Size> field to set the address window size and location. Corresponds to transaction address [31:16].
15:0	Reserved	RSVD 0x0	Reserved

Table 380: PCI Express Window<n+6> Remap Register (n=0–9)
Offset: x10: 0x00081C08, x11: 0x00081C18...x19: 0x00081C98
Offset Formula: 0x00081C08+0x10*n: where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 381: PCI Express Window<n+6> Remap (High) Register (n=0–9)
Offset: x10: 0x00081C0C, x11: 0x00081C1C...x19: 0x00081C9C
Offset Formula: 0x00081C0C+0x10*n: where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap High Address Specifies address bits[63:32] to be driven to the target interface. Only relevant for target interfaces that support a more than 4-GByte address space. When using target interface that do not support a more than 4-GByte address space, this register must be cleared.

A.7.2 PCI Express BAR Control (Port 0)

Table 382: PCI Express BAR1 Control Register
Offset: 0x00081804

Bit	Field	Type/InitVal	Description
31:16	Bar1Size	RW SAR	BAR1 Size BAR sizes range from 64 KByte to 4 GByte in powers of 2. Default value: 03FFF = 1 GByte

Copyright © 2015 Marvell

Table 382: PCI Express BAR1 Control Register (Continued)
Offset: 0x00081804

Bit	Field	Type/InitVal	Description
15:1	Reserved	RSVD 0x0	Reserved
0	Bar1En	RW 0x1	BAR1 Enable 0 = Disable 1 = Enable

Table 383: PCI Express BAR2 Control Register
Offset: 0x00081808

Bit	Field	Type/InitVal	Description
31:16	Bar2Size	RW SAR	BAR2 Size BAR sizes range from 64 KByte to 4 GByte in powers of 2. Default value: 0x0FFF = 256 MByte
15:1	Reserved	RSVD 0x0	Reserved
0	Bar2En	RW 0x1	BAR2 Enable 0 = Disable 1 = Enable

Table 384: PCI Express Expansion ROM BAR Control Register
Offset: 0x0008180C

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved
21:19	ExpROMSz	RW 0x0	Expansion ROM Address Bank Size 0 = 512 KByte 1 = 1024 KByte 2 = 2048 KByte 3 = 4096 KByte
18:1	Reserved	RSVD 0x0	Reserved
0	ExpROMEn	RW 0x0	Expansion ROM BAR Enable 0 = Disable 1 = Enable

Document Classification: Proprietary Information
Copyright © 2015 Marvell

A.7.3 PCI Express Configuration Cycles Generation (Port 0)

Table 385: PCI Express Configuration Address Register
Offset: 0x000818F8

Bit	Field	Type/InitVal	Description
31	ConfigEn	RW 0x0	Configuration Enable Bit
30:28	Reserved	RSVD 0x0	Reserved
27:24	ExtRegNum	RW 0x0	Extended Register Number
23:16	BusNum	RW 0x0	Bus Number
15:11	DevNum	RW 0x0	Device Number
10:8	FunctNum	RW 0x0	Function Number
7:2	RegNum	RW 0x0	Register Number
1:0	Reserved	RSVD 0x0	Reserved

Table 386: PCI Express Configuration Data Register
Offset: 0x000818FC

Bit	Field	Type/InitVal	Description
31:0	ConfigData	RW 0x0	Write access to this register generates a corresponding Configuration TLP on the PCI Express port or an access to the PCI Express port configuration header registers.

A.7.4 PCI Express Configuration Header (Port 0)

Table 387: PCI Express Device and Vendor ID Register
Offset: 0x00080000

Bit	Field	Type/InitVal	Description
31:16	DevID	RO 0x7846	Device ID
15:0	VenID	RO 0x11AB	Vendor ID This field identifies Marvell as the Vendor of the device.

Copyright © 2015 Marvell

Table 388: PCI Express Command and Status Register
Offset: 0x00080004

Bit	Field	Type/InitVal	Description
31	DetParErr	RW1C 0x0	Detected Parity Error This bit is set when the device receives a poisoned TLP. **NOTE:** The bit is set regardless of the state of the <PErrEn> bit in this register. Write 1 to clear.
30	SSysErr	RW1C 0x0	Signalled System Error This bit is set when the device sends an ERR_FATAL or ERR_NONFATAL message. This bit is not set if the <SErrEn> field in this register is de-asserted. Write 1 to clear.
29	RMAbort	RW1C 0x0	Received Master Abort This bit is set when the device, as a requester (master), receives a completion with the status Unsupported Request. Write 1 to clear.
28	RTAbort	RW1C 0x0	Received Target Abort This bit is set when the device, as a requester (master), receives a completion with the status Completer Abort. Write 1 to clear.
27	STarAbort	RW1C 0x0	Signaled Target Abort This bit is set when the device, as a completer (target), completes a transaction as a Completer Abort. Write 1 to clear.
26:25	Reserved	RSVD 0x0	Does not apply to PCI Express. These bits are hardwired to 0.
24	MasDataPerr	RW1C 0x0	Master Data Parity Error This bit is set when the device has poisoned data detected as a requestor (master). Set when the <PErrEn> field (bit[6]) is set and either: Poisoned completion is received for the PCI Express port. or Poisoned TLP is transmitted to the PCI Express port. Write 1 to clear.
23:21	Reserved	RSVD 0x0	These bits are hardwired to 0.
20	CapList	RO 0x1	Capability List Support This bit indicates that the device configuration header includes capability list. This bit is hardwired to 1, since this is always supported in PCI Express.
19	IntStat	RO 0x0	Interrupt Status When set, this bit indicates that an interrupt message is pending internally in the device. 0 = Disable: No interrupt asserted. 1 = Enable: Interrupt asserted.
18:11	Reserved	RSVD 0x0	These bits are hardwired to 0.

Copyright © 2015 Marvell

Table 388: PCI Express Command and Status Register (Continued)
Offset: 0x00080004

Bit	Field	Type/InitVal	Description
10	IntDis	RW 0x0	Interrupt Disable This bit controls the ability of the device to generate interrupt emulation messages. When set, interrupt messages are not generated. 0 = Enable: Interrupt messages enabled. 1 = Disable: Interrupt messages disabled.
9	Reserved	RSVD 0x0	Does not apply to PCI Express. This bit is hardwired to 0.
8	SErrEn	RW 0x0	This bit controls the ability of the device to report fatal and non-fatal errors to the Root Complex. This bit affects both assertion of SSysErr status bit[30] in this register and uncorrectable error message generation. **NOTE:** PCI Express uncorrectable error messages are reported if enabled either through this field or through fields <NFErrRepEn> or <FErrRepEn> in the PCI Express Device Control Status Register. 0 = Disable: SSysErr assertion is disabled. 1 = Enable: SSysErr assertion is enabled. In addition uncorrectable error messages generation is enabled.
7	Reserved	RSVD 0x0	Does not apply to PCI Express. This bit is hardwired to 0.
6	PErrEn	RW 0x0	Parity Error Enable This bit controls the ability of the device to respond to poisoned data errors as a requestor (master) on the PCI Express port. Controls MasDataPerr status bit assertion in the PCMDSTT register. **NOTE:** The setting of this bit does not affect the DetectedPErr status bit. 0 = Disable: MasDataPerr assertion is disabled. 1 = Enable: MasDataPerr assertion is enabled.
5:3	Reserved	RSVD 0x0	Does not apply to PCI Express devices. These bits are hardwired to 0.
2	MasEn	RW 0x0	Master Enable This bit controls the ability of the device to act as a master on the PCI Express port. When set to 0, no memory or I/O read/write request packets are generated to PCI Express. **NOTE:** Messages and completions are transmitted to the PCI Express regardless of the setting of this bit. 0 = Disable: Neither memory nor I/O requests are transmitted to the PCI-E port. 1 = Enable: Memory and I/O requests are transmitted to the PCI-E port.
1	MemEn	RW 0x0	Memory Space Enable Controls the device response to PCI Express memory accesses. 0 = Disable: All Memory accesses from PCI Express are completed as Unsupported Requests. 1 = Enable
0	IOEn	RW 0x0	I/O Space Enable The I/O space is not enabled; therefore, this bit is not functional. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 389: PCI Express Class Code and Revision ID Register
Offset: 0x00080008

Bit	Field	Type/InitVal	Description
31:24	BaseClass	RO SAR	Device Base Class - Memory Controller
23:16	SubClass	RO SAR	Device Sub class--Other Memory Controller
15:8	ProgIF	RO 0x0	Register Level Programming Interface
7:0	RevID	RO 0x0	Device Revision Number

Table 390: PCI Express BIST Header Type and Cache Line Size Register
Offset: 0x0008000C

Bit	Field	Type/InitVal	Description
31	BISTCap	RO 0x0	BIST Capable Bit BIST is not supported. **NOTE:** This bit must be set to 0.
30	BISTAct	RO 0x0	BIST Activate bit BIST is not supported.
29:28	Reserved	RSVD 0x0	Reserved
27:24	BISTComp	RO 0x0	BIST Completion Code 0x0 = BIST passed. Other = BIST failed.
23:16	HeadType	RO 0x0	Device Configuration Header Type The header is a Type 0 single-function configuration header.
15:8	Reserved	RSVD 0x0	Does not apply to PCI Express. These bits are hardwired to 0.
7:0	CacheLine	RW 0x00	Device Cache Line Size **NOTE:**

Table 391: PCI Express BAR0 Internal Register
Offset: 0x00080010

Bit	Field	Type/InitVal	Description
31:20	Base	RW 0xD00	Internal register memory Base Address Indicates 1-MByte address space. Corresponds to address bits[31:20].
19:4	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 391: PCI Express BAR0 Internal Register (Continued)
Offset: 0x00080010

Bit	Field	Type/InitVal	Description
3	Prefetch	RO 0x1	Prefetch Enable Indicates that prefetching is enabled.
2:1	Type	RO 0x2	BAR Type. BAR can be located in 64-bit memory address space.
0	Space	RO 0x0	Memory Space Indicator.

Table 392: PCI Express BAR0 Internal (High) Register
Offset: 0x00080014

Bit	Field	Type/InitVal	Description
31:0	Base	RW 0x0	Base address Corresponds to address bits[63:32].

Table 393: PCI Express BAR1 Register
Offset: 0x00080018

Bit	Field	Type/InitVal	Description
31:16	Base/Reserved	RW 0x0000	Base address Defined according to PCI Express BAR1 Control Register. Indicates 64-KByte to 4-GByte address space. Corresponds to address bits[31:16].
15:4	Reserved	RSVD 0x0	Reserved
3	Prefetch	RO 0x1	Prefetch Enable Indicates that prefetching is enabled.
2:1	Type	RO 0x2	BAR Type BAR can be located in 64-bit memory address space.
0	Space	RO 0x0	Memory Space Indicator

Copyright © 2015 Marvell

Table 394: PCI Express BAR1 (High) Register
Offset: 0x0008001C

Bit	Field	Type/InitVal	Description
31:0	Base	RW 0x0	Base address Corresponds to address bits[63:32].

Table 395: PCI Express BAR2 Register
Offset: 0x00080020

Bit	Field	Type/InitVal	Description
31:16	Base/Reserved	RW 0xF000	Base address Defined according to PCI Express BAR2 Control Register. Indicates 64-KByte up to 4-GByte address space. Corresponds to address bits[31:16].
15:4	Reserved	RSVD 0x0	Reserved
3	Prefetch	RO 0x1	Prefetch Enable Indicates that pre-fetching is enabled.
2:1	Type	RO 0x2	BAR Type BAR can be located in 64-bit memory address space.
0	Space	RO 0x0	Memory Space Indicator

Table 396: PCI Express BAR2 (High) Register
Offset: 0x00080024

Bit	Field	Type/InitVal	Description
31:0	Base	RW 0x0	Base address Corresponds to address bits[63:32].

Table 397: PCI Express Subsystem Device and Vendor ID Register
Offset: 0x0008002C

Bit	Field	Type/InitVal	Description
31:16	SSDevID	RO 0x11AB	Subsystem Device ID Number The default is the Marvell Device ID.
15:0	SSVenID	RO 0x11AB	Subsystem Manufacturer ID Number The default is the Marvell Vendor ID.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 398: PCI Express Expansion ROM BAR Register
Offset: 0x00080030

Bit	Field	Type/InitVal	Description
If expansion ROM is not enabled via the ExpROMEn bit[0] the PCI Express Expansion ROM BAR Control Register, this register is reserved.			
31:22	RomBase	RW 0x0	Expansion ROM Base Address[31:22]
21:19	RomBase/Reserved	RW 0x0	These bits are defined according to field <ExpROMSz> in the PCI Express Expansion ROM BAR Control Register. When ROM Size is: 0 = 512 KByte Space: Bits[21:19] are used as Expansion ROM Base Address[21:19]. 1 = 1 MByte Space: Bits[21:20] are used as Expansion ROM Base Address[21:20], and bit[19] is reserved. 2 = 2 MByte Space: Bits[21] used as Expansion ROM Base Address[21], and bits[20:19] are reserved. 3 = 4 MByte Space: Bits[21:19] are reserved.
18:1	Reserved	RSVD 0x0	Reserved
0	RomEn	RW 0x0	Expansion ROM Enable **NOTE:** The device expansion ROM address space is enabled only if both this bit (RomEn) and <MemEn> in the PCI Express Command and Status Register are set. 0 = Disable 1 = Enable

Table 399: PCI Express Capability List Pointer Register
Offset: 0x00080034

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	CapPtr	RO 0x40	Capability List Pointer The current value in this field points to the PCI Power Management capability set in the PCI Express Power Management Capability Header Register at offset 0x40.

Table 400: PCI Express Interrupt Pin and Line Register
Offset: 0x0008003C

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Does not apply to PCI Express This field is hardwired to 0.
15:8	IntPin	RO 0x01	Indicates that the device is using INTA (Interrupt A) in the interrupt emulation messages.

Copyright © 2015 Marvell

Table 400: PCI Express Interrupt Pin and Line Register (Continued)
Offset: 0x0008003C

Bit	Field	Type/InitVal	Description
7:0	IntLine	RW 0x00	Provides interrupt line routing information.

Table 401: PCI Express Power Management Capability Header Register
Offset: 0x00080040

Bit	Field	Type/InitVal	Description
31:27	PMESup	RO 0x00	This field indicates whether the device supports PM Event generation. The device does not support the PMEn pin.
26	D2Sup	RO 0x1	This bit indicates whether the device supports D2 Power Management state. The device does not support D2 state.
25	D1Sup	RO 0x1	This bit indicates whether the device supports D1 Power Management state. The device does not support D1 state.
24:22	AuxCur	RO 0x0	Auxiliary Current Requirements
21	DSI	RO 0x0	Device Specific Initialization The device does not requires device specific initialization.
20:19	Reserved	RSVD 0x0	Does not apply to PCI Express This field must be hardwired to 0.
18:16	PMCVer	RO 0x3	PCI Power Management Capability Version
15:8	NextPtr	RO 0x50	Next Item Pointer Current value points to MSI capability.
7:0	CapID	RO 0x01	Capability ID Current value identifies the PCI Power Management capability.

Table 402: PCI Express Power Management Control and Status Register
Offset: 0x00080044

Bit	Field	Type/InitVal	Description
31:24	PMData	RO 0x0	State Data This field is used to report the state dependent data requested by the <PMDataSel> field of this register. The value of this field is scaled by the value reported by the <PMDataScale> field of this register.
23:16	Reserved	RSVD 0x0	Does not apply to PCI Express. This field must be hardwired to 0.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 402: PCI Express Power Management Control and Status Register (Continued)
Offset: 0x00080044

Bit	Field	Type/InitVal	Description
15	PMEStat	RW1C 0x0	PME Status **NOTE:** Sticky bit--not initialized by hot reset.
14:13	PMDataScale	RO 0x0	Data Scale Indicates the scaling factor to be used when interpreting the value of the <PMData> field of this register. The read value depends on the setting of the <PMDataSel> field of this register.
12:9	PMDataSel	RW 0x0	Data Select This 4-bit field is used to select which data is to be reported through the <PMDataScale> and <PMData> fields of this register.
8	PME_en	RW 0x0	PM_PME Message Generation Enable **NOTE:** Sticky bit--not initialized by hot reset. 0 = Disable 1 = Enable
7:4	Reserved	RSVD 0x0	Reserved
3	No_Soft_Reset	RO 0x0	When set (1), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. When clear (0), devices do perform an internal reset upon transitioning from D3hot to D0 via software control of the PowerState bits. Configuration Context is lost when performing the soft reset. Upon transition from the D3hot to the D0 state, full reinitialization sequence is needed to return the device to D0 Initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 uninitialized with only PME context preserved if PME is supported and enabled.
2	Reserved	RSVD 0x0	Reserved
1:0	PMState	RW 0x0	Power State This field controls the Power Management state of the device. The device supports all Power Management states. 0 = D0 1 = D1 2 = D2 3 = D3

Copyright © 2015 Marvell

Table 403: PCI Express MSI Message Control Register
Offset: 0x00080050

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23	Addr64	RO 0x1	64-bit Addressing Capable This field indicates whether the device is capable of generating a 64-bit message address. 0 = NotCapable 1 = Capable
22:20	MultiEn	RW 0x0	Multiple Messages Enable The number of messages the system allocates to the device. This number must be smaller or equal to the value that is in the <MultiCap> field.
19:17	MultiCap	RO 0x0	Multiple Message Capable The device is capable of driving a single message.
16	MSIEn	RW 0x0	MSI Enable Controls the device interrupt signaling mechanism. 0 = Disable: Interrupts are signaled through the interrupt emulation messages. 1 = Enable: Interrupts are signaled through MSI mechanism.
15:8	NextPtr	RO 0x60	Next Item Pointer The current value of this field points to PCI Express capability.
7:0	CapID	RO 0x5	Capability ID The current value of this field identifies the PCI MSI capability.

Table 404: PCI Express MSI Message Address Register
Offset: 0x00080054

Bit	Field	Type/InitVal	Description
31:2	MSIAddr	RW 0x0	Message Address This field corresponds to Address[31:2] of the MSI MWr TLP.
1:0	Reserved	RSVD 0x0	Reserved

Table 405: PCI Express MSI Message Address (High) Register
Offset: 0x00080058

Bit	Field	Type/InitVal	Description
31:0	MSIAddrh	RW 0x0	Message Upper Address This field corresponds to Address[63:32] of the MSI MWr TLP.

Copyright © 2015 Marvell

Table 406: PCI Express MSI Message Data Register
Offset: 0x0008005C

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	MSIData	RW 0x0	Message Data **NOTE:**

Table 407: PCI Express Capability Register
Offset: 0x00080060

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:25	IntMsgNum	RO 0x0	Interrupt Message Number
24	SlotImp	RO 0x0	Slot Implemented
23:20	DevType	RO SAR	Device/Port Type 1 = EndPoint: Legacy PCI Express Endpoint 4 = RootComplex: Root Port of PCI Express Root Complex
19:16	CapVer	RO 0x2	Capability Version This field indicates the PCI Express Base spec 2.0 version of the PCI Express capability.
15:8	NextPtr	RO 0x0	Next Item Pointer The current value of this field points to the end of the capability list (NULL).
7:0	CapID	RO 0x10	Capability ID The current value of this field identifies the PCI PE capability.

Table 408: PCI Express Device Capabilities Register
Offset: 0x00080064

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved

Table 408: PCI Express Device Capabilities Register (Continued)
Offset: 0x00080064

Bit	Field	Type/InitVal	Description
28	FuncLevelRstCpb	RO 0x0	Function Level Reset Capability. A value of 1b indicates the Function supports the optional Function Level Reset mechanism.
27:26	CapSPLScl	RO 0x0	Captured Slot Power Limit Scale
25:18	CapSPLVal	RO 0x0	Captured Slot Power Limit Value
17:16	Reserved	RSVD 0x0	Reserved
15	RuleBaseErrorRep	RO 0x1	Role-Based Error Reporting => This bit, when set, indicates that the device implements the functionality originally defined in the Error Reporting ECN for PCI Express Base Specification, Revision 1.0a, and later incorporated into PCI Express Base Specification, Revision 1.1. This bit must be set by all devices conforming to the ECN, PCI Express Base Specification, Revision 1.1., or subsequent PCI Express Base Specification revisions.
14	Reserved	RO 0x0	Reserved
13	Reserved	RO 0x0	Reserved
12	Reserved	RO 0x0	Reserved
11:9	EPL1AccLat	RO 0x0	Endpoint L1 Acceptable Latency This field indicates the amount of latency that can be absorbed by the device due to the transition from L1 to L0. The current value specifies less than 1 second. 000b Maximum of 1 µs 001b Maximum of 2 µs 010b Maximum of 4 µs 011b Maximum of 8 µs 100b Maximum of 16 µs 101b Maximum of 32 µs 110b Maximum of 64 µs 111b No limit

Copyright © 2015 Marvell

Table 408: PCI Express Device Capabilities Register (Continued)
Offset: 0x00080064

Bit	Field	Type/InitVal	Description
8:6	EPL0sAccLat	RO 0x2	Endpoint L0s Acceptable Latency This field indicates the amount of latency that can be absorbed by the device due to the transition from L0s to L0. 0 = Under64ns: Less than 64 ns. 1 = 64to128ns: Less than 128 ns. 2 = 128to256ns: Less than 256 ns. 3 = 256to512ns: Less than 512 ns. 4 = 512nsto1us: Less than 1 us. 5 = 1to2us: Less than 2 us. 6 = 2to4us: Less than 4 us. 7 = Above4us: No limit
5:3	Reserved	RO 0x0	Reserved These bits are hardwired to 0.
2:0	MaxPldSizeSup	RO 0x0	Maximum Payload Size Supported 128B MPS support. 0 = 128

Table 409: PCI Express Device Control Status Register
Offset: 0x00080068

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved
21	TransPend	RO 0x0	Transactions Pending Indicates that the device has issued Non-Posted requests that have not been completed. 0 = Completed: All NP requests have been completed or terminated by the Completion Timeout Mechanism. 1 = Uncompleted: Not all NP requests have been completed or terminated.
20	Reserved	RSVD 0x0	Reserved
19	URDet	RW1C 0x0	Unsupported Request Detected Indicates that the device received an unsupported request. Write 1 to clear.
18	FErrDet	RW1C 0x0	Fatal Error Detected Indicates the status of the fatal errors detected by the device. Write 1 to clear.
17	NFErrDet	RW1C 0x0	Non-Fatal Error Detected Indicates the status of the non-fatal errors detected by the device. Write 1 to clear.
16	CorErrDet	RW1C 0x0	Correctable Error Detected Indicates the status of the correctable errors detected by the device. Write 1 to clear.
15	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 409: PCI Express Device Control Status Register (Continued)
Offset: 0x00080068

Bit	Field	Type/InitVal	Description
14:12	MaxRdRqSz	RW 0x2	Maximum Read Request Size Limits the device maximum read request size as a requestor (master). Other = Reserved 0 = 128B 1 = 256B 2 = 512B 3 = 1 KByte 4 = 2 KByte 5 = 4 KByte
11	EnNS	RO 0x0	Enable No Snoop The device never sets the No Snoop attribute in transactions it initiates as a requester. Hardwired to 0x0. 0 = Disable 1 = Enable
10	Reserved	RSVD 0x0	Reserved
9:8	Reserved	RO 0x0	Reserved These bits are hardwired to 0.
7:5	MaxPldSz	RW 0x0	Maximum Payload Size The maximum payload size supported is 128B (refer to bit <MaxPldSizeSup> in the PCI Express Device Capabilities Register). Other values = Reserved 0 = 128B: 0
4	EnRO	RO 0x0	Enable Relaxed Ordering The device never sets the Relaxed Ordering attribute in transactions it initiates as a requester. Hardwired to 0x0. 0 = Disable 1 = Enable
3	URRepEn	RW 0x0	Unsupported Request (UR) Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated, therefore always write 0x0. **NOTE:** UR related error messages are still enabled when URRepEn=0, if <SErrEn> bit in PCI Express Command and Status Register is set. 0 = Disabled: UR related error messages are masked. Status bit is not masked. 1 = Enabled: UR related error messages enabled.

Copyright © 2015 Marvell

Table 409: PCI Express Device Control Status Register (Continued)
Offset: 0x00080068

Bit	Field	Type/InitVal	Description
2	FErrRepEn	RW 0x0	Fatal Error Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated; therefore, always write 0x0. **NOTE:** ERR_FATAL error messages are still enabled when this field is 0, if the <SErrEn> bit in PCI Express Command and Status Register is set. 0 = Disable: ERR_FATAL error messages are masked. Status bit is still affected. 1 = Enable: ERR_FATAL error messages enabled.
1	NFErrRepEn	RW 0x0	Non-Fatal Error Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated, therefore always write 0x0. **NOTE:** ERR_NONFATAL error messages are still enabled when this field is 0, if the <SErrEn> bit in PCI Express Command and Status Register is set. 0 = Disable: ERR_NONFATAL error messages are masked. Status bit is not masked. 1 = Enable: ERR_NONFATAL error messages enabled.
0	CorErrRepEn	RW 0x0	Correctable Error Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated; therefore, always write 0x0. **NOTE:** ERR_NONFATAL error messages are still enabled when this field is 0, if the <SErrEn> bit in the PCI Express Command and Status Register is set. 0 = Disable: ERR_COR error messages are masked. Status bit is not masked. 1 = Enable: ERR_COR error messages enabled.

Table 410: PCI Express Link Capabilities Register
Offset: 0x0008006C

Bit	Field	Type/InitVal	Description
31:24	PortNum	RO 0x0	In Endpoint mode, indicates the PCI Express port number as was advertised by the Root Complex during the link training process.
23:19	Reserved	RSVD 0x0	Reserved

Table 410: PCI Express Link Capabilities Register (Continued)
Offset: 0x0008006C

Bit	Field	Type/InitVal	Description
18	ClockPowerMng	RO 0x1	Clock Power Management A value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the clock request (CLKREQ#) mechanism when the link is in the L1 and L2/3 Ready link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these link states. This capability is applicable only in form factors that support clock request (CLKREQ#) capability. For a multi-function device, each function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the multifunction device indicate a 1b in this bit.
17:15	L1ExtLat	RO 0x7	L1 Exit Latency This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this port requires to complete transition from L1 to L0. Defined encodings are: 000b = Less than 1µs 001b = 1 µs to less than 2 µs 010b = 2 µs to less than 4 µs 011b = 4 µs to less than 8 µs 100b = 8 µs to less than 16 µs 101b = 16 µs to less than 32 µs 110b = 32 µs-64 µs 111b = More than 64 µs **NOTE:** The exit latencies can be influenced by the PCI Express reference clock configuration, depending upon whether a component uses a common or separate reference clock.
14:12	L0sExtLat	RO 0x2	L0s Exit Latency The time required by the device to transition its Rx lanes from L0s to L0. 0 = Under64ns: Less than 64 ns 1 = 64to128ns: 64 ns-128 ns 2 = 128to256ns: 128 ns-256 ns 3 = 256to512ns: 256 ns-512 ns 4 = 512nsto1µs: 512 ns-1 µs 5 = 1to2µs: 1 µs-2 µs 6 = 2to4µs: 2 us-4 us 7 = Reserved
11:10	AspmSup	RO 0x3	Active State Link PM Support The current value identifies L0s and L1 Supported. 1 = L0s_Entry_Supported 3 = L0s_and_L1_Supported
9:4	MaxLnkWdth	RO SAR	Maximum Link Width supported by the port. The legal values are 1 for X1 link and 4 for X4 link.

Copyright © 2015 Marvell

Table 410: PCI Express Link Capabilities Register (Continued)
Offset: 0x0008006C

Bit	Field	Type/InitVal	Description
3:0	MaxLinkSpd	RO SAR	Maximum Link Speed The current value identifies the 2.5 Gbps link. 1 = 2.5G Support: 2.5 GTb/s Link speed supported. 2 = 5G Support: 5.0 GT/s and 2.5 GT/s Link speeds supported.

Table 411: PCI Express Link Control Status Register
Offset: 0x00080070

Bit	Field	Type/InitVal	Description
31	PCIeLinkAutoBwStt	RW1C 0x0	Link Autonomous Bandwidth Status This bit is set by hardware to indicate that hardware has autonomously changed the link speed or width, without the port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable link operation. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was indicated as an autonomous change.
30	PCIeLinkBwMngStt	RW1C 0x0	Link Bandwidth Management Status – This bit is set by hardware to indicate that either of the following has occurred without the port transitioning through DL_Down status: - A Link retraining has completed following a write of 1b to the Retrain Link bit. **NOTE:** This bit is set following any write of 1b to the Retrain Link bit, including when the link is in the process of retraining for some other reason. - Hardware has changed the link speed or width to attempt to correct unreliable link operation, either through an LTSSM timeout or a higher level process. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change.
29	Reserved	RSVD 0x0	Reserved
28	SltClkCfg	RO 0x1	Slot Clock Configuration 0 = Independent: The device uses an independent clock, irrespective of the presence of a reference clock on the connector. 1 = Reference: The device uses the reference clock that the platform provides.
27	LnkTrn	RO 0x0	Link Training This bit is not applicable and reserved for Endpoints.
26	Reserved	RO 0x0	Reserved
25:20	NegLnkWdth	RO 0x0	Negotiated Link Width This field indicates the negotiated link width. This field is initialized by the hardware. The value in this field is undefined when the link is not up. 1 = X1 4 = X4

Table 411: PCI Express Link Control Status Register (Continued)
Offset: 0x00080070

Bit	Field	Type/InitVal	Description
19:16	LnkSpd	RO 0x1	Link Speed This field indicates the negotiated link speed. 1 = 2.5G: 2.5 GT/s PCI Express Link 2 = 5G: 5.0 GT/s PCI Express Link
15:10	Reserved	RSVD 0x0	Reserved
9	HW_autonomous_width_disable	RW 0x0	Hardware Autonomous Width Disable When set, this bit disables hardware from changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width.
8	EnClkPwrMng	RW 0x0	Enable Clock Power Management Applicable only for form factors that support a Clock Request (CLKREQ#) mechanism, this enable functions as follows: 0b = Clock power management is disabled and device must hold CLKREQ# signal low. 1b = When this bit is set to 1, the device is permitted to use CLKREQ# signal to power manage link clock according to protocol defined in appropriate form factor specification. Default value of this field is 0b. Components that do not support Clock Power Management (as indicated by a 0b value in the Clock Power Management bit of the Link Capabilities register) must hardwire this bit to 0b.
7	ExtdSnc	RW 0x0	Extended Sync When set, this bit forces extended transmission of 4096 FTS ordered sets followed by a single skip ordered set in exit from L0s and extra (1024) TS1 at exit from L1. **NOTE:** This bit is used for test and measurement.
6	CmnClkCfg	RW 0x0	Common Clock Configuration When set by the software, this bit indicates that both devices on the link use a distributed common reference clock.
5	RetrnLnk	RW 0x0	Retrain Link This bit forces the device to initiate link retraining. Always returns 0 when read. **NOTE:** If configured as an Endpoint, this field is reserved and has no effect.
4	LnkDis	RW 0x0	Link Disable **NOTE:** If configured as an Endpoint, this field is reserved and has no effect. Activation procedure: 1. Set this bit to trigger link disable. 2. Poll <DLDown> de-assertion (PCI Express Status Register, bit 0) ensure the link is disabled. 3. Clear the bit to exit to detect and enable the link again.
3	RCB	RW 0x0	Read Completion Boundary **NOTE:** This bit has no effect on the device behavior. Completions are always returned with 128B read completion boundary. 0 = 64B: 64 Bytes 1 = 128B: 128 Bytes

Document Classification: Proprietary Information

Table 411: PCI Express Link Control Status Register (Continued)
Offset: 0x00080070

Bit	Field	Type/InitVal	Description
2	Reserved	RSVD 0x0	Reserved
1:0	AspmCnt	RW 0x0	Active State Link PM Control Controls the level of active state PM supported on the link. 0 = Disable 1 = L0 Support: L0s entry supported. 2 = Reserved 3 = L0L1 Support: L0s and L1 entry supported.

Table 412: PCI Express Device Capabilities 2 Register
Offset: 0x00080084

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4	CplTODisSup	RO 0x0	Completion Timeout Disable Supported A value of 1b indicates support for the Completion Timeout Disable mechanism.
3:0	CplTORangesSup	RO 0x0	Completion Timeout Ranges Supported This field indicates device function support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. Four time value ranges are defined: Range A: 50 us to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4s Range D: 4 s to 64 s Bits are set according to the table below to show the timeout value ranges supported. 0000b = Completion Timeout programming not supported. The function must implement a timeout value in the range 50 us to 50 ms. 0001b = Range A 0010b = Range B 0011b = Ranges A and B 0110b = Ranges B and C 0111b = Ranges A, B, and C 1110b = Ranges B, C and D 1111b = Ranges A, B, C, and D All other values are reserved.

Table 413: PCI Express Device Control Status 2 Register
Offset: 0x00080088

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved

Table 413: PCI Express Device Control Status 2 Register (Continued)
Offset: 0x00080088

Bit	Field	Type/InitVal	Description
4	CplTODis	RW 0x0	Completion Timeout Disable When set, this bit disables the Completion Timeout mechanism. Software is permitted to set or clear this bit at any time. When set, the Completion Timeout detection mechanism is disabled. If there are outstanding requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding requests. If this is done, it is permitted to base the start time for each request on either the time this bit was cleared or the time each request was issued.
3:0	CplTOValue	RW 0x0	Completion Timeout Value Defined encoding: 0000b Default range: 50 us to 50 ms **NOTE:** Marvell strongly recommends that the Completion Timeout mechanism not expire in less than 10 ms. Values available if Range A (50 us to 10 ms) programmability range is supported: 0001b = 50 us to 100 us 0010b = 1 ms to 10 ms Values available if Range B (10 ms to 250 ms) programmability range is supported: 0101b = 16 ms to 55 ms 0110b = 65 ms to 210 ms Values available if Range C (250 ms to 4 s) programmability range is supported: 1001b = 260 ms to 900 ms 1010b = 1s to 3.5s Values available if the Range D (4 s to 64 s) programmability range is supported: 1101b = 4s to 13s 1110b = 17s to 64s Values not defined above are reserved.

Table 414: PCI Express Link Control Status 2 Register
Offset: 0x00080090

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	CurrentDeemphasis Level	RO 0x0	Current De-emphasis Level When the Link operates at 5 GT/s speed, this bit reflects the level of de-emphasis. Values: 1b -3.5 dB 0b -6 dB The value in this bit is undefined when the link operates at 2.5GT/s speed.
15:13	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 414: PCI Express Link Control Status 2 Register (Continued)
Offset: 0x00080090

Bit	Field	Type/InitVal	Description
12	ComplianceDeemphasis	RW 0x0	Compliance De-emphasis This bit sets the de-emphasis level in Polling. Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Values: 1b -3.5 dB 0b -6 dB This bit is intended for debug, compliance testing purposes. System firmware and software are allowed to modify this bit only during debug or compliance testing.
11	ComplianceSOS	RW 0x0	Compliance SOS When set to 1b, the LTSSM is required to send SKP Ordered Sets periodically in between the (modified) compliance patterns.
10	EnterModifiedCompliance	RW 0x0	Enter Modified Compliance When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. This fiel is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value.
9:7	TransmitMargin	RO 0x0	Transmit Margin This field controls the value of the nonde-emphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see the relevant section in the Functional Specifications about how the transmitter voltage level is determined in various states). Values: a) 000b = Normal operating range b) 001b = 800-1200 mV for full swing and 400-700 mV for half-swing c) 010b - (n-1) = Values must be monotonic with a non-zero slope. The value of n must be greater than 3 and less than 7. At least two of these must be below the normal operating range of n: 200-400 mV for full-swing and 100-200 mV for half-swing d) (n) - 111b = Reserved **NOTE:** This register can be written to, although it is appears as RO. This register is intended for debug, compliance testing purposes only. System firmware and software are allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.
6	SelectDeemphasis	RO 0x0	Selectable De-emphasis When the link is operating at 5.0 GT/s speed, this bit selects the level of de-emphasis for an Upstream component. Values: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream ports of switches. This field can be changed from the Mbus.

Table 414: PCI Express Link Control Status 2 Register (Continued)
Offset: 0x00080090

Bit	Field	Type/InitVal	Description
5	HardwareAutoSpeed Dis	RO 0x0	Hardware Autonomous Speed Disable When set, this bit disables hardware from changing the link speed for device specific reasons other than attempting to correct unreliable link operation by reducing link speed. Initial transition to the highest supported common link speed is not blocked by this bit. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. PCIe does not implement this feature.
4	EnterCompliance	RW 0x0	Enter Compliance Software is permitted to force a link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link.
3:0	TargetLinkSpeed	RW 0x2	Target Link Speed - For Downstream Ports, this field sets an upper limit on link operational speed by restricting the values advertised by the Upstream component in its training sequences. Defined values: 0001b = 2.5 GT/s Target Link Speed 0010b = 5.0 GT/s Target Link Speed All other values are reserved. If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, the result is undefined. The default value of this field is the highest link speed supported by the component (as reported in the Supported Link Speeds field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value. For both Upstream and Downstream ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode.

Table 415: PCI Express Advanced Error Report Header Register
Offset: 0x00080100

Bit	Field	Type/InitVal	Description
31:20	NextPtr	RO 0x0	Next Item Pointer This field indicates the last item in the extended capabilities linked list.
19:16	CapVer	RO 0x1	Capability Version
15:0	PECapID	RO 0x1	Extended Capability ID The current value of this field identifies the Advanced Error Reporting capability.

Copyright © 2015 Marvell

Table 416: PCI Express Uncorrectable Error Status Register
Offset: 0x00080104

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset. All fields in this register (except for reserved fields) are SC. Write 1 to clear. A write of 0 has no effect.			
31:21	Reserved	RSVD 0x0	Reserved
20	URErr	RW1C 0x0	Unsupported Request Error Status **NOTE:** Sticky bit--not initialized by hot reset.
19	Reserved	RSVD 0x0	Reserved
18	MalfTlpErr	RW1C 0x0	Malformed TLP Status **NOTE:** Sticky bit--not initialized by hot reset.
17	Reserved	RW1C 0x0	Reserved **NOTE:** Sticky bit--not initialized by hot reset. **NOTE:**
16	UnexpCmpErr	RW1C 0x0	Unexpected Completion Status **NOTE:** Sticky bit--not initialized by hot reset.
15	CAErr	RW1C 0x0	Completer Abort Status **NOTE:** Sticky bit--not initialized by hot reset.
14	CmpTOErr	RW1C 0x0	Completion Timeout Status **NOTE:** Sticky bit--not initialized by hot reset.
13	Reserved	RSVD 0x0	Reserved **NOTE:** Sticky bit--not initialized by hot reset. **NOTE:**
12	RPsnTlpErr	RW1C 0x0	Poisoned TLP Status **NOTE:** Hot sticky bit--not initialized by hot reset.
11:5	Reserved	RSVD 0x0	Reserved
4	DLPrtErr	RW1C 0x0	Data Link Protocol Error Status **NOTE:** Hot sticky bit--not initialized by hot reset.
3:0	Reserved	RSVD 0x0	Reserved

Table 417: PCI Express Uncorrectable Error Mask Register
Offset: 0x00080108

Bit	Field	Type/InitVal	Description
All fields in this register are sticky not initialized or modified by hot reset.			
31:21	Reserved	RSVD 0x0	Reserved

Table 417: PCI Express Uncorrectable Error Mask Register (Continued)
Offset: 0x00080108

Bit	Field	Type/InitVal	Description
20	URErrMsk	RW 0x0	Unsupported Request Error Mask
19	Reserved	RSVD 0x0	Reserved.
18	MalfTlpErrMsk	RW 0x0	Malformed TLP Mask
17	Reserved	RW 0x0	Reserved
16	UnexpCmpErrMsk	RW 0x0	Unexpected Completion Mask
15	CAErrMsk	RW 0x0	Completer Abort Mask
14	CmpTOErrMsk	RW 0x0	Completion Timeout Mask
13	Reserved	RSVD 0x0	Reserved
12	RPsnTlpErrMsk	RW 0x0	Poisoned TLP Error Mask
11:5	Reserved	RSVD 0x0	Reserved
4	DLPrtErrMsk	RW 0x0	Data Link Protocol Error Mask When an error is indicated in the PCI Express Uncorrectable Error Status Register and the corresponding bit is set: - The header is not logged in the Header Log Register. - The First Error Pointer is not updated. - An error message is not generated. Status bit is set regardless of the mask setting. 0 = NotMasked 1 = Masked
3:0	Reserved	RSVD 0x0	Reserved

Table 418: PCI Express Uncorrectable Error Severity Register
Offset: 0x0008010C

Bit	Field	Type/InitVal	Description
All fields in this register are hot sticky not initialized or modified by reset.			
31:21	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 418: PCI Express Uncorrectable Error Severity Register (Continued)
Offset: 0x0008010C

Bit	Field	Type/InitVal	Description
20	URErrSev	RW 0x0	Unsupported Request Error Severity
19	Reserved	RSVD 0x0	Reserved
18	MalfTlpErrSev	RW 0x1	Malformed TLP Severity
17	Reserved	RW 0x1	Reserved
16	UnexpCmpErrSev	RW 0x0	Unexpected Completion Severity
15	CAErrSev	RW 0x0	Completer Abort Severity
14	CmpTOErrSev	RW 0x0	Completion Timeout Severity
13	Reserved	RSVD 0x0	Reserved
12	RPsnTlpErrSev	RW 0x0	Poisoned TLP Error Severity
11:5	Reserved	RSVD 0x0	Reserved
4	DLPrtErrSev	RW 0x1	Data Link Protocol Error Severity Controls the severity indication of the Uncorrectable errors. Each bit controls the error type of the corresponding bit in the PCI Express Uncorrectable Error Status Register. 0 = NonFatal: Error type is Non-Fatal. 1 = Fatal: Error type is Fatal.
3:0	Reserved	RSVD 0x0	Reserved

Table 419: PCI Express Correctable Error Status Register
Offset: 0x00080110

Bit	Field	Type/InitVal	Description
All fields in this register are sticky not initialized or modified by hot reset. All fields in this register (except for reserved fields) are RW1C write 1 to clear.			
31:14	Reserved	RSVD 0x0	Reserved
13	AdvNonFatalErr	RW1C 0x0	Advisory Non-Fatal Error Status
12	RplyTOErr	RW1C 0x0	Replay Timer Timeout Status

Copyright © 2015 Marvell

Table 419: PCI Express Correctable Error Status Register (Continued)
Offset: 0x00080110

Bit	Field	Type/InitVal	Description
11:9	Reserved	RSVD 0x0	Reserved
8	RplyRllovrErr	RW1C 0x0	Replay Number Rollover Status
7	BadDllpErr	RW1C 0x0	Bad DLLP Status
6	BadTlpErr	RW1C 0x0	Bad TLP Status
5:1	Reserved	RSVD 0x0	Reserved
0	RcvErr	RW1C 0x0	Receiver Error Status **NOTE:** When set, this bit indicates that a Receiver error has occurred.

Table 420: PCI Express Correctable Error Mask Register
Offset: 0x00080114

Bit	Field	Type/InitVal	Description
All fields in this register are sticky not initialized or modified by hot reset.			
31:14	Reserved	RSVD 0x0	Reserved
13	AdvNonFatalMsk	RW 0x1	Advisory Non-Fatal Error Mask This bit is set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.
12	RplyTOMsk	RW 0x0	Replay Timer Timeout Mask
11:9	Reserved	RSVD 0x0	Reserved
8	RplyRllovrMsk	RW 0x0	Replay Number Rollover Mask
7	BadDllpErrMsk	RW 0x0	Bad DLLP Mask
6	BadTlpMsk	RW 0x0	Bad TLP Mask
5:1	Reserved	RSVD 0x0	Reserved
0	RcvMsk	RW 0x0	Receiver Error Mask If set, an error message is not generated upon occurrence of a Receiver error. 0 = NotMasked 1 = Masked

Document Classification: Proprietary Information

Table 421: PCI Express Advanced Error Capability and Control Register
Offset: 0x00080118

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved **NOTE:** Bits [5] and [7] are Read Only and are hardwired to 0.
4:0	FrstErrPtr	RO 0x0	First Error Pointer This field reports the bit position of the first error reported in the PCI Express Uncorrectable Error Status Register. This field locks upon receipt of the first uncorrectable error that is not masked. It remains locked until software clears it by writing 1 to the corresponding status bit. Upon receipt of the next uncorrectable error that is not masked, the field locks again until cleared as described above. This lock and clear process continues to repeat itself. **NOTE:** The bits in this field are sticky bits--they are not initialized or modified by reset. 4 = DLP: Data Link Protocol Error 12 = TLP: Poisoned TLP Error 13 = FCP: Flow Control Protocol Error 14 = CT: Completion Timeout Error 15 = CAS: Completer Abort Status 16 = UCE: Unexpected Completion Error 17 = ROE: Receiver Overflow Error 18 = Mutant TLP: Malformed TLP Error 19 = Reserved 20 = URE: Unsupported Request Error Other = Reserved

Table 422: PCI Express Header Log First DWORD Register
Offset: 0x0008011C

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog1DW	RO 0x0	Header Log First DWORD Logs the header of the first error reported in the PCI Express Uncorrectable Error Status Register (PEUEST). This field locks upon receipt of the first uncorrectable error that is not masked. It remains locked until software clears it by writing 1 to the corresponding status bit. Upon receipt of the next uncorrectable error that is not masked, the field locks again until cleared as described above. This lock and clear process continues to repeat itself. **NOTE:**

Table 423: PCI Express Header Log Second DWORD Register
Offset: 0x00080120

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog2DW	RO 0x0	Header Log Second DWORD **NOTE:**

Table 424: PCI Express Header Log Third DWORD Register
Offset: 0x00080124

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog3DW	RO 0x0	Header Log Third DWORD **NOTE:**

Table 425: PCI Express Header Log Fourth DWORD Register
Offset: 0x00080128

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog4DW	RO 0x0	Header Log Fourth DWORD **NOTE:** All fields in this register are Hot sticky bit -- not initialized or modified by hot reset.

Table 426: PCI Express Root Error Command Register
Offset: 0x0008012C

Bit	Field	Type/InitVal	Description
31:3	Reserved	RO 0x0	Reserved
2	Fatal Error Reporting Enable	RW 0x0	Fatal Error Reporting Enable When set, this bit enables the generation of an interrupt when a fatal error is reported by any of the functions in the hierarchy associated with this Root Port.
1	Non-Fatal Error Reporting Enable	RW 0x0	Non-Fatal Error Reporting Enable When set, this bit enables the generation of an interrupt when a non-fatal error is reported by any of the functions in the hierarchy associated with this Root Port.
0	Correctable Error Reporting Enable	RW 0x0	Correctable Error Reporting Enable When set, this bit enables the generation of an interrupt when a correctable error is reported by any of the functions in the hierarchy associated with this Root Port.

Table 427: PCI Express Root Error Status Register
Offset: 0x00080130

Bit	Field	Type/InitVal	Description
31:27	Advanced Error Interrupt Message Number	RO 0x0	Advanced Error Interrupt Message Number This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this capability.
26:7	Reserved	RO 0x0	RsvdZ

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 427: PCI Express Root Error Status Register (Continued)
Offset: 0x00080130

Bit	Field	Type/InitVal	Description
6	Fatal Error Messages Received	RW1C 0x0	Fatal Error Messages Received Set when one or more fatal uncorrectable error messages have been received.
5	Non-Fatal Error Messages Received	RW1C 0x0	Non-Fatal Error Messages Received Set when one or more non-fatal uncorrectable error messages have been received.
4	First Uncorrectable Fatal	RW1C 0x0	First Uncorrectable Fatal Set when the first uncorrectable error message received is for a fatal error.
3	Multiple ERR_FATAL/NONFATAL Received	RW1C 0x0	Multiple ERR_FATAL/NONFATAL Received Set when either a fatal or a non-fatal error is received and ERR_FATAL/NONFATAL Received is already set.
2	ERR_FATAL/NONFATAL Received	RW1C 0x0	ERR_FATAL/NONFATAL Received Set when either a fatal or a non-fatal error message is received and this bit is not already set.
1	Multiple ERR_COR Received	RW1C 0x0	Multiple ERR_COR Received Set when a correctable error message is received and ERR_COR Received is already set.
0	ERR_COR Received	RW1C 0x0	ERR_COR Received Set when a correctable error message is received and this bit is not already set.

Table 428: PCI Express Error Source Identification Register
Offset: 0x00080134

Bit	Field	Type/InitVal	Description
31:16	ERR_FATAL/NONFATAL Source Identification	ROC 0x0	ERR_FATAL/NONFATAL Source Identification Loaded with the Requester ID indicated in the received ERR_FATAL or ERR_NONFATAL message when the ERR_FATAL/NONFATAL Received bit is not already set.
15:0	ERR_COR Source Identification	ROC 0x0	ERR_COR Source Identification Loaded with the Requester ID indicated in the received ERR_COR Message when the ERR_COR Received bit is not already set.

A.7.5 PCI Express Control and Status (Port 0)

Table 429: PCI Express Control Register
Offset: 0x00081A00

Bit	Field	Type/InitVal	Description
31	CRS Enable	RW 0x0	When this bit is set, all configuration access returns with status=<CRS>. 0 = Disable 1 = Enable

Table 429: PCI Express Control Register (Continued)
Offset: 0x00081A00

Bit	Field	Type/InitVal	Description
30	Conf Training Disable	RW 0x1	MAC Training Disable
29	Reserved	RSVD 0x0	Reserved
28	ConfMove2DetectFromModCompl	RW 0x0	If asserted, causes link restart while in Polling.MOD.COMPL state.
27	ConfMstrDis Scrmb	RW 0x0	Master Disable Scrambling When set, a scrambling disable command is transmitted downstream, causing the scrambling to be disabled on both sides of the link. **NOTE:** This should be programmed before the start of link initialization.
26	ConfMstrLb	RW 0x0	Master Loopback When set, a loopback command is transmitted downstream, causing the downstream device to transmit back the traffic that it receives. This bit can be set only if bits[24] of this register and <LnkDis> in the PCI Express Link Control Status Register are cleared. **NOTE:** Use this bit only when working in Root Complex mode.
25	Compliance Receive	RW 0x0	Compliance Receive When set, a compliance command (Bit 4 of Symbol 5 in TS1 OS) is asserted in TS1 OS, causing the device connected to move to Polling.Compliance state.
24	ConfMstrHot Reset	RW 0x0	Master Hot-Reset When set, a hot-reset command is transmitted downstream, causing the downstream PCI Express hierarchy to enter a hot-reset cycle. This bit can be set only if LnkDis in the PCI Express Link Control Status Register and bit[26] of this register are cleared. Activation procedure: 1. Set this bit to trigger a hot-reset cycle. 2. Poll DLDown de-assertion (PCI Express Status Register, bit 0) to ensure hot-reset cycle is done. 3. Clear this bit to exit to detect and re-establish the PCI-Express link. This bit should be used only when working in Root Complex mode.
23:16	Reserved	RW 0x14	Reserved Must write 0x14. **NOTE:**
15:13	Reserved	RSVD 0x0	Reserved
12	Reserved	RW 0x0	Reserved
11	Reserved	RW 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 429: PCI Express Control Register (Continued)
Offset: 0x00081A00

Bit	Field	Type/InitVal	Description
10	conf_auto_speed_change	RW 0x0	Auto Speed change. when set , link will issue link speed change to the max link speed possibel. 0x0 do not initiate a change. 0x1 do initiate a change.
9:8	Reserved_9_8	RW SAR	Reserved Please do not change default value at any time. The default value = 0x2
7:3	Reserved	RSVD 0x5	Reserved
2	CfgMapTo MemEn	RW 0x1	Configuration Header Mapping to Memory Space Enable When enabled, the configuration header registers can be accessed directly through the memory space. Access is enabled both from the internal bus and from the PCI Express port. 0 = Disable: Mapping disabled. 1 = Enable: Mapping enabled.
1	ConfRoot Complex	RW SAR	PCI Express Device Type- this field is not in use in PCIe ports 0 and 1 and should be ignored 0 = Endpoint: Endpoint device (Downstream device / Upstream link). 1 = Root: Root Complex device (Upstream device / Downstream link).
0	Reserved	RW 0x0	Reserved

Table 430: PCI Express Status Register
Offset: 0x00081A04

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27	PCIeSlvDis Scrmb	RO 0x0	Slave Disable Scrambling Indication This field sets when the opposite device on the PCI Express port acts as a disable scrambling master, and when a scrambling disabled indication is received.
26	PCIeSlvLb	RO 0x0	Slave Loopback Indication This field sets when the opposite device on the PCI Express port acts as a loopback master, and when a loopback indication is received.
25	PCIeSlvDisLink	RO 0x0	Slave Disable Link Indication This field sets when the opposite device on the PCI Express port acts as a disable link master, and when a link disabled indication is received.
24	PCIeSlvHot Reset	RO 0x0	Slave Hot Reset Indication This field sets when the opposite device on the PCI Express port acts as a hot-reset master, and a when hot-reset indication is received.

Copyright © 2015 Marvell

Table 430: PCI Express Status Register (Continued)
Offset: 0x00081A04

Bit	Field	Type/InitVal	Description
23:21	Reserved	RSVD 0x0	Reserved
20:16	PCIeDevNum	RW 0x0	Device Number Indication This field is used in the RequesterID field of the transmitted TLPs. In Endpoint mode, the field updates whenever a CfgWr access is received.
15:8	PCIeBusNum	RW 0x0	Bus Number Indication This field is used in the RequesterID field of the transmitted TLPs. In Endpoint mode, the field updates whenever a CfgWr access is received.
7:5	Reserved	RSVD 0x0	Reserved
4:1	Reserved	RSVD 0x0	Reserved
0	DLDown	RO 0x0	DL_Down Indication 0 = Active: DL is up. 1 = Inactive: DL is down.

Table 431: PCI Express Spare Register
Offset: 0x00081A08

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0xFFFF0007	Reserved

Table 432: PCI Express Root Complex Set Slot Power Limit (SSPL) Register
Offset: 0x00081A0C

Bit	Field	Type/InitVal	Description
31:17	Reserved	RO 0x0	Reserved
16	SsplMsgEnable	RW 0x0	SSPL Message Enable 0 = Disable: No SSPL message is send. 1 = Enable: SSPL message is send when the PCI Express is set to Root Complex and: 1) DI-Down -> DI-Up or 2) There is a Write to the <SlotPowerLimitValue> or <SlotPowerLimitScale> fields.
15:10	Reserved	RO 0x0	Reserved

Document Classification: Proprietary Information

Table 432: PCI Express Root Complex Set Slot Power Limit (SSPL) Register (Continued)
Offset: 0x00081A0C

Bit	Field	Type/InitVal	Description
9:8	SlotPowerLimitScale	RW 0x0	Slot Power Limit Scale Specifies the scale used for the Slot Power Limit Value (in the PCI Express Interface section see the sub-section "Messages"). Range of Values: 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x Writes to this register when the PCI Express interface is set to Root Complex also cause the port to send the Set Slot Power Limit message.
7:0	SlotPowerLimitValue	RW 0x0	Slot Power Limit Value In combination with the <SlotPowerLimitScale> value, specifies the upper limit on power supplied by the slot. The power limit (in watts) calculated by multiplying the value in this field by the value in the <SlotPowerLimitScale> field. Writes to this register when the PCI Express interface is set to Root Complex also cause the port to send the Set Slot Power Limit message.

Table 433: PCI Express Completion Timeout Register
Offset: 0x00081A10

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	CPL To Range Offset	RW 0x0	Completion Range Offset This field is used to select a value from the range indicated in the <CplTOValue> field in the PCI Express Control Status 2 register. For example, the <CplTOValue> field provides a default range: 50 us to 50 ms and this register indicates whether the value is close to the minimum or maximum value in this range. 0 = Min Value: Minimum value in the selected range. 1 = Max Value: Maximum value in the selected range.
15:0	Reserved	RW 0x2710	Reserved

Table 434: PCI Express Root Complex Power Management Event (PME) Register
Offset: 0x00081A14

Bit	Field	Type/InitVal	Description
31:18	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 434: PCI Express Root Complex Power Management Event (PME) Register (Continued)
Offset: 0x00081A14

Bit	Field	Type/InitVal	Description
17	PME Pending	RO 0x0	PME Pending This read-only bit indicates that another PME is pending when the PME Status bit is set. When the PME Status bit is cleared by software, the PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending.
16	PME Status	RO 0x0	PME Status This bit indicates that PME was asserted by the requestor ID indicated in the PME Requestor ID field. This bit can be cleared by a write of 0x1 to the <RcvPmPme> field in the PCI Express Interrupt Cause register (0x1900[28]).
15:0	PME Requester ID	RO 0x0	PME Requestor ID This field indicates the PCI requestor ID of the last locked PME requestor. When PME_Status=0x0, the value in the PME_Requester_ID is regard as reserved.

Table 435: PCI Express Power Management Register
Offset: 0x00081A18

Bit	Field	Type/InitVal	Description
31:17	Reserved	RW 0x0	Reserved
16	Ref_clk_off_en	RW 0x0	The system enables the hardware to kill the REFCLK and Core Clock. This input is also a condition for gating the PCI Express unit.
15:6	Reserved	RSVD 0x0	Reserved
5	Send Turn Off Msg	RW 0x0	Host Request PCI Express unit to send Turn Off message.
4	Send Turn Off Ack Msg	RW 0x0	Endpoint Ready for Turn Off and ready to send Turn Off ack message and start sending DLLP Enter L2/3.
3:2	Reserved	RSVD 0x0	Reserved
1	L1 ASPM Ack	RW 0x1	Only for Root Complex. The system allows acknowledgment of an L1 ASPM request by an Endpoint. 0 = SendAspmNackTlp: Stop L1-ASPM flow. 1 = SendAspmAckDllp: Continue L1-ASPM flow.
0	L1 ASPM En	RW 0x0	Only for Endpoint. Enables the hardware to initialize the L1 ASPM process. 0 = IDLE: No l1-Aspm flow (as EndPoint) 1 = StartL1AspmFlow

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 436: PCI Express Flow Control Register
Offset: 0x00081A20

Bit	Field	Type/InitVal	Description
31	Disable_UpdateFC_TO	RW 0x0	Disable Update Flow Control due to 30 usec timeout. 0 = EnableUpdateFC: There will be at least one Update Flow Control each 30 usec. 1 = DisableUpdateFC: There will be no Update Flow Control due to 30 usec timer expiration.
30:24	Reserved	RW 0x0	Reserved
23:16	ConfChInitFc	RW 0x0	Completion Headers Flow Control Credit Initial Value Infinite.
15:8	ConfNphInitFc	RW SAR	Non-Posted Headers Flow Control Credit Initial Value The default value = 0x8
7:0	ConfPhInitFc	RW SAR	Posted Headers Flow Control Credit Initial Value The default value = 0x4

Table 437: PCI Express Replay Timeout (TO) Timer Register
Offset: 0x00081A24

Bit	Field	Type/InitVal	Description
31:10	Reserved	RSVD 0x0	Reserved
9	Disable Replay TO	RW 0x0	Disable Replay TO 1 = No Replay TO: There will be no Replay TO.
8	Use Change Replay TO	RW 0x0	Use Change Replay TO 0 = Fixed: The design value is taken from the fixed initial value. 1 = Bit[7:0] Value: The design value is taken from the <Change Replay TO> bits[7:0].
7:0	Change Replay TO	RW 0x0	Change Replay TO value used when <Use Change Replay TO> is asserted.

Table 438: PCI Express Ack Latency Timer Register
Offset: 0x00081A28

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8	Use Change Ack TO	RW 0x0	Use Change Ack TO 0 => The design value is taken from fixed init-val. 1 => The design value is taken from change_ack_to[7:0]. 0 = Fixed: The design value is taken from fixed initial value. 1 = Bit[7:0] Value: The design value is taken from the <Use Change Ack TO> field bits[7:0].

Copyright © 2015 Marvell

Table 438: PCI Express Ack Latency Timer Register (Continued)
Offset: 0x00081A28

Bit	Field	Type/InitVal	Description
7:0	Change Ack TO	RW 0x0	The value used for ACT_TO when the <Use Change Ack TO> field is asserted.

Table 439: PCI Express Secondary Bus Register
Offset: 0x00081A2C

Bit	Field	Type/InitVal	Description
31:9	Reserved	RO 0x0	
8	Secondary Bus Number Enable	RW 0x0	When 1, the CFG TLP TYPE is defined using Secondary Bus number. Otherwise the Primary bus number is used.
7:0	Secondary Bus Number	RW 0x0	The Secondary Bus number field is used to decide on type on CFG request. If the Destination Bus number equals the Secondary Bus number then type 0; otherwise type 1. This field is used if the <Secondary Bus Number Enable> field is 1'b1; otherwise the Primary Bus is used for the type calculation.

A.7.6 PCI Express Debug (Port 0)

Table 440: PCI Express Debug Control Register
Offset: 0x00081A60

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x3D	Reserved
21	DisHotResetRegRst	RW 0x1	Disable PCIe Register File reset upon hot reset. **NOTE:** Sticky bit -- not initialized by reset. If set to 0x0 (reset is enabled), Regfile initialization is necessary again, if it overwrites the PCIe configurations. 0 = Disable: Mask Disable 1 = Enable: Mask Enable
20	Reserved	RSVD 0x0	Reserved
19	DisLinkRestartRegRst	RW 0x1	Disable PCIe Register File reset upon Link restart (as a result of link disable prosedure or link instability) Note: Sticky bit - not initialized by reset 0 = Disable: Mask Disable 1 = Enable: Mask Enable
18	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 440: PCI Express Debug Control Register (Continued)
Offset: 0x00081A60

Bit	Field	Type/InitVal	Description
17	ConfMskHotReset	RW 0x1	Mask Hot Reset from PCIe Reset Output **NOTE:** Sticky bit -- not initialized by reset. 0 = Disable: Mask Disable 1 = Enable: Mask Enable
16	ConfMskLnkRestart	RW 0x1	Mask Reset propagation to PCIE Reset Output in case of link restart event (as a result of link disable prosedure or link instability) Note: Sticky bit - not initialized by reset 0 = Disable: Mask Disable 1 = Enable: Mask Enable
15:0	Reserved	RSVD 0xF0C0	Reserved

Table 441: PCI Express Debug MAC Status 1 Register
Offset: 0x00081A84

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:24	TrnCntlBits	RO 0x0	Training control status bits sent on TX: Bit 27 - Scramble disable Bit 26 - Master loopback Bit 25 - Link disable Bit 24 - Hot reset
23:0	Reserved	RSVD 0x0	Reserved

Table 442: PCI Express TL Control Register
Offset: 0x00081AB0

Bit	Field	Type/InitVal	Description
31:14	Reserved	RSVD 0x0	Reserved
13:10	PDedicateBufferSize	RW 0x3	Indicates the number of entries used for posted tlps allocation. The maximum number for X1 mode is 5 , for X4 mode is 11. The number of CPL entries is calculated as 6 - <PDedicateBufferSize> for X1 mode and 12 - <PDedicateBufferSize> for X4 mode.
9	PCplDedicateAlloc	RW 0x0	If asserted, indicates that the dynamic buffer allocation is disabled and the dedicated buffer sizes are used for completion (CPL) and posted (P) buffers. 0 = Dynamic allocation: Dynamic allocation is enabled. 1 = Dedicated allocation: CPL and P tlps each have their own resources.

Copyright © 2015 Marvell

Table 442: PCI Express TL Control Register (Continued)
Offset: 0x00081AB0

Bit	Field	Type/InitVal	Description
8	UpdateDedicatedBufferAlloc	RW 0x0	To latch the allocation of Posted and Completion buffers from the <PDedicateBufferSize> field of current register, the user must write 1'b1 to this field. Otherwise, the design will assume that the size of Completion and Posted buffers are equal. To update the size of buffers again, without initiating a fundamental reset, write 1'b0 to this field and then 1'b1 again.
7	Reserved	RSVD 0x0	Reserved
6	Reserved	RW 0x0	Reserved
5	Reserved	RW 0x0	Reserved
4	Reserved	RW 0x0	Reserved
3	TxNpPushDis	RW 0x0	Tx Transaction Ordering Control (Non Posted vs. Posted) 0 = Disable 1 = Enable: Tx non posted may bypass posted.
2	TxCmplPushDis	RW 0x0	Tx Transaction Ordering Control (Completion vs. Posted) 0 = Disable 1 = Enable: Tx completion may bypass posted.
1	RxNpPushDis	RW 0x0	Rx Transaction Ordering Control (Non Posted vs. Posted) 0 = Disable 1 = Enable: Rx non posted may bypass posted.
0	RxCmplPushDis	RW 0x0	Rx Transaction Ordering Control (Completion vs. Posted) 0 = Disable 1 = Enable: Rx completion may bypass posted.

A.7.7 PCI Express Interrupt (Port 0)

Table 443: PCI Express Interrupt Cause Register
Offset: 0x00081900

Bit	Field	Type/InitVal	Description
All bits, except bits[27:24], are RW0C only. A cause bit sets upon an event occurrence. A write of 0 clears the bit. A write of 1 has no effect. Bits[24:27] are set and cleared upon reception of interrupt emulation messages.			
31	RcvMsi	RW0C 0x0	RC received MemWr that hit the MSI attributes.
30	Reserved	RW0C 0x0	Reserved
29	RcvTurnOff	RW0C 0x0	If PCIe is Endpoint, RcvTurnOffMsg: Receive Turn Off MSG from the RC. If PCIe is RootComplex, Ready4TurnOff: RC notifies the Host that the Turn Off Process is completed (whether normally or by a timer).

Document Classification: Proprietary Information

Table 443: PCI Express Interrupt Cause Register (Continued)
Offset: 0x00081900

Bit	Field	Type/InitVal	Description
28	RcvPmPme	RW0C 0x0	Received PM_PME message.
27	RcvIntD	RO 0x0	IntDn status Reflects IntD Interrupt message emulation status. Set when IntD_Assert message received. Cleared when IntD_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
26	RcvIntC	RO 0x0	IntC status Reflects IntC Interrupt message emulation status. Set when IntC_Assert message received. Cleared when IntC_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
25	RcvIntB	RO 0x0	IntB status Reflects IntB Interrupt message emulation status. Set when IntB_Assert message received. Cleared when IntB_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
24	RcvIntA	RO 0x0	IntA status Reflects IntA Interrupt message emulation status. Set when IntA_Assert message received. Cleared when IntA_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
23	PCIeLinkFail	RW0C 0x0	Link Failure indication PCI Express link dropped from active state (L0, L0s or L1) to Detect state due to link errors. **NOTE:** When dropping to Detect via Hot Reset, Disable Link or Loopback states, the interrupt is not asserted. Sticky bit--not initialized by reset.
22	PCIeSlvLb	RW0C 0x0	Slave Loopback Indication The bit sets when the opposite device on the PCI Express port is acting as a loopback master, and loopback mode was entered. **NOTE:** Sticky bit--not initialized by reset.
21	PCIeSlvDisLink	RW0C 0x0	Slave Disable Link Indication The bit sets when the opposite device on the PCI Express port is acting as a disable link master, and link was disabled. **NOTE:** Sticky bit--not initialized by reset.

Table 443: PCI Express Interrupt Cause Register (Continued)
Offset: 0x00081900

Bit	Field	Type/InitVal	Description
20	PCIeSlvHot Reset	RW0C 0x0	Received Hot Reset Indication The bit sets when a hot-reset indication is received from the opposite device on the PCI Express port. **NOTE:** Sticky bit--not initialized by reset.
19	RcvCRS	RW0C 0x0	Received CRS completion status. A downstream PCI Express device can respond to a configuration request with CRS (Configuration Request Retry Status) if it is not ready yet to serve the request. RcvCRS interrupt is set when such a completion status is received.
18	RcvErrCor	RW0C 0x0	Received ERR_COR message. Set when a downstream device detected the error and sent an error message upstream. Relevant for Root Complex only.
17	RcvErrNonFatal	RW0C 0x0	Received ERR_NONFATAL message. Set when a downstream device detected the error and sent an error message upstream. Relevant for Root Complex only.
16	RcvErrFatal	RW0C 0x0	Received ERR_FATAL message. Set when a downstream device detected the error and sent an error message upstream. Relevant for Root Complex only.
15	RcvUrCaErr	RW0C 0x0	Received either UR or CA completion status.
14	FlowCtrlProtocol	RW0C 0x0	Flow Control Protocol Error Set when there is no FC DLLP for more than 200 usec.
13	Reserved	RW0C 0x0	Reserved
12	BIST	RW0C 0x0	PCI Express BIST Enable BIST is not supported.
11	DstateChange	RW0C 0x0	Dstate Change Indication Any change in the Dstate asserts this bit. **NOTE:** This bit is relevant only for Endpoint.
10	FErrDet	RW0C 0x0	Fatal Error Detected Indicates status of Fatal errors detected by the device.
9	NFErrDet	RW0C 0x0	Non-Fatal Error Detected Indicates status of Non-Fatal errors detected by the device.
8	CorErrDet	RW0C 0x0	Correctable Error Detected Indicates status of correctable errors detected by the device.
7	TxRamParErr	RW0C 0x0	Tx RAM Parity Error
6	RxRamParErr	RW0C 0x0	Rx RAM Parity Error

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 443: PCI Express Interrupt Cause Register (Continued)
Offset: 0x00081900

Bit	Field	Type/InitVal	Description
5	Reserved	RW0C 0x0	Reserved
4	HitDfltWinErr	RW0C 0x0	Hit Default Window Error
3	ErrWrToReg	RW0C 0x0	Erroneous write attempt to internal register. Set when an erroneous write request to PCI Express internal register is received, either from the PCI Express (EP set) or from the internal bus (bit[64] set).
2	Reserved	RW0C 0x0	Reserved
1	MDis	RW0C 0x0	Attempt to generate a PCI transaction while master is disabled.
0	TxReqInDldownErr	RW0C 0x0	Receive TxReq while PCI Express is in Dl-Down.

Table 444: PCI Express Interrupt Mask Register
Offset: 0x00081910

Bit	Field	Type/InitVal	Description
31:24	Mask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the interrupt.
23:20	Mask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the interrupt.
19:0	Mask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the interrupt.

A.7.8 PCI Express Mbus Control (Port 0)

Table 445: PCI Express Mbus Adapter Control Register
Offset: 0x000818D0

Bit	Field	Type/InitVal	Description
31:21	Reserved	RSVD 0x0	Reserved
20	TxDPPropEn	RW 0x1	Tx Data Poisoning Error Propagation Enable 0 = Enable 1 = Disable

Table 445: PCI Express Mbus Adapter Control Register (Continued)
Offset: 0x000818D0

Bit	Field	Type/InitVal	Description
19	RxDPPropEn	RW 0x1	Rx Data Poisoning Error Propagation Enable 0 = Disable 1 = Enable
18:6	Reserved	RSVD 0x0	Reserved
5	Reserved	RW 0x0	Reserved
4	Reserved	RW 0x0	Reserved
3:2	Reserved	RW 0x0	Reserved
1	Reserved	RW 0x0	Reserved
0	Reserved	RW 0x0	Reserved

Table 446: PCI Express Mbus Arbiter Control Register (Low)
Offset: 0x000818E0

Bit	Field	Type/InitVal	Description
31:28	Arb7	RW 0x7	Slice 7 of arbiter.
27:24	Arb6	RW 0x6	Slice 6 of arbiter.
23:20	Arb5	RW 0x5	Slice 5 of arbiter.
19:16	Arb4	RW 0x4	Slice 4 of arbiter.
15:12	Arb3	RW 0x3	Slice 3 of arbiter.
11:8	Arb2	RW 0x2	Slice 2 of arbiter.
7:4	Arb1	RW 0x1	Slice 1 of arbiter.
3:0	Arb0	RW 0x0	Slice 0 of arbiter.

Copyright © 2015 Marvell

Table 447: PCI Express Mbus Arbiter Control Register (High)
Offset: 0x000818E4

Bit	Field	Type/InitVal	Description
31:28	Arb15	RW 0xf	Slice 15 of arbiter.
27:24	Arb14	RW 0xe	Slice 14 of arbiter.
23:20	Arb13	RW 0xd	Slice 13 of arbiter.
19:16	Arb12	RW 0xc	Slice 12 of arbiter.
15:12	Arb11	RW 0xb	Slice 11 of arbiter.
11:8	Arb10	RW 0xa	Slice 10 of arbiter.
7:4	Arb9	RW 0x9	Slice 9 of arbiter.
3:0	Arb8	RW 0x8	Slice 8 of arbiter.

Table 448: PCI Express Mbus Arbiter Timeout Register
Offset: 0x000818E8

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	TimeoutEn	RW 0x1	Mbus Arbiter Timeout Timer Enable 0 = Enable 1 = Disable
15:8	Reserved	RSVD 0x0	Reserved
7:0	Timeout	RW 0xFF	Mbus Arbiter Timeout Preset Value **NOTE:** Must keep the value of 0xFF as long as the Timeout counter is not enabled.

Copyright © 2015 Marvell

A.8 PCI Express 2.0 Ports 1-3 Registers

The following table provides a summarized list of all registers that belong to the PCI Express 2.0 Ports 1-3, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers

Register Name	Offset	Table and Page
PCI Express Configuration Cycles Generation (Ports 1-3)		
PCI Express Configuration Address Register (p=1–3)	pcie_port_num1: 0x000418F8, pcie_port_num2: 0x000458F8, pcie_port_num3: 0x000498F8	Table 450, p. 634
PCI Express Configuration Data Register (p=1–3)	pcie_port_num1: 0x000418FC, pcie_port_num2: 0x000458FC, pcie_port_num3: 0x000498FC	Table 451, p. 634
PCI Express Configuration Header (Ports 1-3)		
PCI Express Device and Vendor ID Register (p=1–3)	pcie_port_num1: 0x00040000, pcie_port_num2: 0x00044000, pcie_port_num3: 0x00048000	Table 452, p. 634
PCI Express Command and Status Register (p=1–3)	pcie_port_num1: 0x00040004, pcie_port_num2: 0x00044004, pcie_port_num3: 0x00048004	Table 453, p. 635
PCI Express Class Code and Revision ID Register (p=1–3)	pcie_port_num1: 0x00040008, pcie_port_num2: 0x00044008, pcie_port_num3: 0x00048008	Table 454, p. 637
PCI Express BIST Header Type and Cache Line Size Register (p=1–3)	pcie_port_num1: 0x0004000C, pcie_port_num2: 0x0004400C, pcie_port_num3: 0x0004800C	Table 455, p. 637
PCI Express BAR0 Internal Register (p=1–3)	pcie_port_num1: 0x00040010, pcie_port_num2: 0x00044010, pcie_port_num3: 0x00048010	Table 456, p. 637
PCI Express BAR0 Internal (High) Register (p=1–3)	pcie_port_num1: 0x00040014, pcie_port_num2: 0x00044014, pcie_port_num3: 0x00048014	Table 457, p. 638
PCI Express BAR1 Register (p=1–3)	pcie_port_num1: 0x00040018, pcie_port_num2: 0x00044018, pcie_port_num3: 0x00048018	Table 458, p. 638
PCI Express BAR1 (High) Register (p=1–3)	pcie_port_num1: 0x0004001C, pcie_port_num2: 0x0004401C, pcie_port_num3: 0x0004801C	Table 459, p. 639
PCI Express BAR2 Register (p=1–3)	pcie_port_num1: 0x00040020, pcie_port_num2: 0x00044020, pcie_port_num3: 0x00048020	Table 460, p. 639
PCI Express BAR2 (High) Register (p=1–3)	pcie_port_num1: 0x00040024, pcie_port_num2: 0x00044024, pcie_port_num3: 0x00048024	Table 461, p. 639
PCI Express Subsystem Device and Vendor ID Register (p=1–3)	pcie_port_num1: 0x0004002C, pcie_port_num2: 0x0004402C, pcie_port_num3: 0x0004802C	Table 462, p. 639

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Expansion ROM BAR Register (p=1–3)	pcie_port_num1: 0x00040030, pcie_port_num2: 0x00044030, pcie_port_num3: 0x00048030	Table 463, p. 640
PCI Express Capability List Pointer Register (p=1–3)	pcie_port_num1: 0x00040034, pcie_port_num2: 0x00044034, pcie_port_num3: 0x00048034	Table 464, p. 640
PCI Express Interrupt Pin and Line Register (p=1–3)	pcie_port_num1: 0x0004003C, pcie_port_num2: 0x0004403C, pcie_port_num3: 0x0004803C	Table 465, p. 641
PCI Express Power Management Capability Header Register (p=1–3)	pcie_port_num1: 0x00040040, pcie_port_num2: 0x00044040, pcie_port_num3: 0x00048040	Table 466, p. 641
PCI Express Power Management Control and Status Register (p=1–3)	pcie_port_num1: 0x00040044, pcie_port_num2: 0x00044044, pcie_port_num3: 0x00048044	Table 467, p. 642
PCI Express MSI Message Control Register (p=1–3)	pcie_port_num1: 0x00040050, pcie_port_num2: 0x00044050, pcie_port_num3: 0x00048050	Table 468, p. 643
PCI Express MSI Message Address Register (p=1–3)	pcie_port_num1: 0x00040054, pcie_port_num2: 0x00044054, pcie_port_num3: 0x00048054	Table 469, p. 643
PCI Express MSI Message Address (High) Register (p=1–3)	pcie_port_num1: 0x00040058, pcie_port_num2: 0x00044058, pcie_port_num3: 0x00048058	Table 470, p. 644
PCI Express MSI Message Data Register (p=1–3)	pcie_port_num1: 0x0004005C, pcie_port_num2: 0x0004405C, pcie_port_num3: 0x0004805C	Table 471, p. 644
PCI Express Capability Register (p=1–3)	pcie_port_num1: 0x00040060, pcie_port_num2: 0x00044060, pcie_port_num3: 0x00048060	Table 472, p. 644
PCI Express Device Capabilities Register (p=1–3)	pcie_port_num1: 0x00040064, pcie_port_num2: 0x00044064, pcie_port_num3: 0x00048064	Table 473, p. 645
PCI Express Device Control Status Register (p=1–3)	pcie_port_num1: 0x00040068, pcie_port_num2: 0x00044068, pcie_port_num3: 0x00048068	Table 474, p. 646
PCI Express Link Capabilities Register (p=1–3)	pcie_port_num1: 0x0004006C, pcie_port_num2: 0x0004406C, pcie_port_num3: 0x0004806C	Table 475, p. 648
PCI Express Link Control Status Register (p=1–3)	pcie_port_num1: 0x00040070, pcie_port_num2: 0x00044070, pcie_port_num3: 0x00048070	Table 476, p. 650
PCI Express Device Capabilities 2 Register (p=1–3)	pcie_port_num1: 0x00040084, pcie_port_num2: 0x00044084, pcie_port_num3: 0x00048084	Table 477, p. 652
PCI Express Device Control Status 2 Register (p=1–3)	pcie_port_num1: 0x00040088, pcie_port_num2: 0x00044088, pcie_port_num3: 0x00048088	Table 478, p. 653
PCI Express Link Control Status 2 Register (p=1–3)	pcie_port_num1: 0x00040090, pcie_port_num2: 0x00044090, pcie_port_num3: 0x00048090	Table 479, p. 653

Copyright © 2015 Marvell

Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Advanced Error Report Header Register (p=1–3)	pcie_port_num1: 0x00040100, pcie_port_num2: 0x00044100, pcie_port_num3: 0x00048100	Table 480, p. 656
PCI Express Uncorrectable Error Status Register (p=1–3)	pcie_port_num1: 0x00040104, pcie_port_num2: 0x00044104, pcie_port_num3: 0x00048104	Table 481, p. 656
PCI Express Uncorrectable Error Mask Register (p=1–3)	pcie_port_num1: 0x00040108, pcie_port_num2: 0x00044108, pcie_port_num3: 0x00048108	Table 482, p. 657
PCI Express Uncorrectable Error Severity Register (p=1–3)	pcie_port_num1: 0x0004010C, pcie_port_num2: 0x0004410C, pcie_port_num3: 0x0004810C	Table 483, p. 658
PCI Express Correctable Error Status Register (p=1–3)	pcie_port_num1: 0x00040110, pcie_port_num2: 0x00044110, pcie_port_num3: 0x00048110	Table 484, p. 659
PCI Express Correctable Error Mask Register (p=1–3)	pcie_port_num1: 0x00040114, pcie_port_num2: 0x00044114, pcie_port_num3: 0x00048114	Table 485, p. 660
PCI Express Advanced Error Capability and Control Register (p=1–3)	pcie_port_num1: 0x00040118, pcie_port_num2: 0x00044118, pcie_port_num3: 0x00048118	Table 486, p. 660
PCI Express Header Log First DWORD Register (p=1–3)	pcie_port_num1: 0x0004011C, pcie_port_num2: 0x0004411C, pcie_port_num3: 0x0004811C	Table 487, p. 661
PCI Express Header Log Second DWORD Register (p=1–3)	pcie_port_num1: 0x00040120, pcie_port_num2: 0x00044120, pcie_port_num3: 0x00048120	Table 488, p. 661
PCI Express Header Log Third DWORD Register (p=1–3)	pcie_port_num1: 0x00040124, pcie_port_num2: 0x00044124, pcie_port_num3: 0x00048124	Table 489, p. 662
PCI Express Header Log Fourth DWORD Register (p=1–3)	pcie_port_num1: 0x00040128, pcie_port_num2: 0x00044128, pcie_port_num3: 0x00048128	Table 490, p. 662
PCI Express Root Error Command Register (p=1–3)	pcie_port_num1: 0x0004012C, pcie_port_num2: 0x0004412C, pcie_port_num3: 0x0004812C	Table 491, p. 662
PCI Express Root Error Status Register (p=1–3)	pcie_port_num1: 0x00040130, pcie_port_num2: 0x00044130, pcie_port_num3: 0x00048130	Table 492, p. 663
PCI Express Error Source Identification Register (p=1–3)	pcie_port_num1: 0x00040134, pcie_port_num2: 0x00044134, pcie_port_num3: 0x00048134	Table 493, p. 663
PCI Express Address Window Control (Ports 1-3)		
PCI Express Window0 Control Register (p=1–3)	pcie_port_num1: 0x00041820, pcie_port_num2: 0x00045820, pcie_port_num3: 0x00049820	Table 494, p. 664
PCI Express Window0 Base Register (p=1–3)	pcie_port_num1: 0x00041824, pcie_port_num2: 0x00045824, pcie_port_num3: 0x00049824	Table 495, p. 664

Document Classification: Proprietary Information

Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Window0 Remap Register (p=1–3)	pcie_port_num1: 0x0004182C, pcie_port_num2: 0x0004582C, pcie_port_num3: 0x0004982C	Table 496, p. 665
PCI Express Window1 Control Register (p=1–3)	pcie_port_num1: 0x00041830, pcie_port_num2: 0x00045830, pcie_port_num3: 0x00049830	Table 497, p. 665
PCI Express Window1 Base Register (p=1–3)	pcie_port_num1: 0x00041834, pcie_port_num2: 0x00045834, pcie_port_num3: 0x00049834	Table 498, p. 666
PCI Express Window1 Remap Register (p=1–3)	pcie_port_num1: 0x0004183C, pcie_port_num2: 0x0004583C, pcie_port_num3: 0x0004983C	Table 499, p. 666
PCI Express Window2 Control Register (p=1–3)	pcie_port_num1: 0x00041840, pcie_port_num2: 0x00045840, pcie_port_num3: 0x00049840	Table 500, p. 666
PCI Express Window2 Base Register (p=1–3)	pcie_port_num1: 0x00041844, pcie_port_num2: 0x00045844, pcie_port_num3: 0x00049844	Table 501, p. 667
PCI Express Window2 Remap Register (p=1–3)	pcie_port_num1: 0x0004184C, pcie_port_num2: 0x0004584C, pcie_port_num3: 0x0004984C	Table 502, p. 667
PCI Express Window3 Control Register (p=1–3)	pcie_port_num1: 0x00041850, pcie_port_num2: 0x00045850, pcie_port_num3: 0x00049850	Table 503, p. 667
PCI Express Window3 Base Register (p=1–3)	pcie_port_num1: 0x00041854, pcie_port_num2: 0x00045854, pcie_port_num3: 0x00049854	Table 504, p. 668
PCI Express Window3 Remap Register (p=1–3)	pcie_port_num1: 0x0004185C, pcie_port_num2: 0x0004585C, pcie_port_num3: 0x0004985C	Table 505, p. 668
PCI Express Window4 Control Register (p=1–3)	pcie_port_num1: 0x00041860, pcie_port_num2: 0x00045860, pcie_port_num3: 0x00049860	Table 506, p. 669
PCI Express Window4 Base Register (p=1–3)	pcie_port_num1: 0x00041864, pcie_port_num2: 0x00045864, pcie_port_num3: 0x00049864	Table 507, p. 669
PCI Express Window4 Remap Register (p=1–3)	pcie_port_num1: 0x0004186C, pcie_port_num2: 0x0004586C, pcie_port_num3: 0x0004986C	Table 508, p. 669
PCI Express Window4 Remap (High) Register (p=1–3)	pcie_port_num1: 0x00041870, pcie_port_num2: 0x00045870, pcie_port_num3: 0x00049870	Table 509, p. 670
PCI Express Window5 Control Register (p=1–3)	pcie_port_num1: 0x00041880, pcie_port_num2: 0x00045880, pcie_port_num3: 0x00049880	Table 510, p. 670
PCI Express Window5 Base Register (p=1–3)	pcie_port_num1: 0x00041884, pcie_port_num2: 0x00045884, pcie_port_num3: 0x00049884	Table 511, p. 671
PCI Express Window5 Remap Register (p=1–3)	pcie_port_num1: 0x0004188C, pcie_port_num2: 0x0004588C, pcie_port_num3: 0x0004988C	Table 512, p. 671

Copyright © 2015 Marvell

Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Window5 Remap (High) Register (p=1–3)	pcie_port_num1: 0x00041890, pcie_port_num2: 0x00045890, pcie_port_num3: 0x00049890	Table 513, p. 671
PCI Express Default Window Control Register (p=1–3)	pcie_port_num1: 0x000418B0, pcie_port_num2: 0x000458B0, pcie_port_num3: 0x000498B0	Table 514, p. 671
PCI Express Expansion ROM Window Control Register (p=1–3)	pcie_port_num1: 0x000418C0, pcie_port_num2: 0x000458C0, pcie_port_num3: 0x000498C0	Table 515, p. 672
PCI Express Expansion ROM Window Remap Register (p=1–3)	pcie_port_num1: 0x000418C4, pcie_port_num2: 0x000458C4, pcie_port_num3: 0x000498C4	Table 516, p. 672
PCI Express Window<n+6> Control Register (n=0–9, p=1–3)	pcie_port_num1x10: 0x00041C00, pcie_port_num1x11: 0x00041C10...pcie_port_num3x19: 0x00049C90	Table 517, p. 673
PCI Express Window<n+6> Base Register (n=0–9, p=1–3)	pcie_port_num1x10: 0x00041C04, pcie_port_num1x11: 0x00041C14...pcie_port_num3x19: 0x00049C94	Table 518, p. 673
PCI Express Window<n+6> Remap Register (n=0–9, p=1–3)	pcie_port_num1x10: 0x00041C08, pcie_port_num1x11: 0x00041C18...pcie_port_num3x19: 0x00049C98	Table 519, p. 674
PCI Express Window<n+6> Remap (High) Register (n=0–9, p=1–3)	pcie_port_num1x10: 0x00041C0C, pcie_port_num1x11: 0x00041C1C...pcie_port_num3x19: 0x00049C9C	Table 520, p. 674
PCI Express Mbus Control (Ports 1-3)		
PCI Express Mbus Adapter Control Register (p=1–3)	pcie_port_num1: 0x000418D0, pcie_port_num2: 0x000458D0, pcie_port_num3: 0x000498D0	Table 521, p. 674
PCI Express Mbus Arbiter Control Register (Low) (p=1–3)	pcie_port_num1: 0x000418E0, pcie_port_num2: 0x000458E0, pcie_port_num3: 0x000498E0	Table 522, p. 675
PCI Express Mbus Arbiter Control Register (High) (p=1–3)	pcie_port_num1: 0x000418E4, pcie_port_num2: 0x000458E4, pcie_port_num3: 0x000498E4	Table 523, p. 675
PCI Express Mbus Arbiter Timeout Register (p=1–3)	pcie_port_num1: 0x000418E8, pcie_port_num2: 0x000458E8, pcie_port_num3: 0x000498E8	Table 524, p. 676
PCI Express Control and Status (Ports 1-3)		
PCI Express Control Register (p=1–3)	pcie_port_num1: 0x00041A00, pcie_port_num2: 0x00045A00, pcie_port_num3: 0x00049A00	Table 525, p. 676
PCI Express Status Register (p=1–3)	pcie_port_num1: 0x00041A04, pcie_port_num2: 0x00045A04, pcie_port_num3: 0x00049A04	Table 526, p. 678
PCI Express Spare Register (p=1–3)	pcie_port_num1: 0x00041A08, pcie_port_num2: 0x00045A08, pcie_port_num3: 0x00049A08	Table 527, p. 679
PCI Express Root Complex Set Slot Power Limit (SSPL) Register (p=1–3)	pcie_port_num1: 0x00041A0C, pcie_port_num2: 0x00045A0C, pcie_port_num3: 0x00049A0C	Table 528, p. 679

Document Classification: Proprietary Information

Table 449: Summary Map Table for the PCI Express 2.0 Ports 1-3 Registers (Continued)

Register Name	Offset	Table and Page
PCI Express Completion Timeout Register (p=1–3)	pcie_port_num1: 0x00041A10, pcie_port_num2: 0x00045A10, pcie_port_num3: 0x00049A10	Table 529, p. 680
PCI Express Root Complex Power Management Event (PME) Register (p=1–3)	pcie_port_num1: 0x00041A14, pcie_port_num2: 0x00045A14, pcie_port_num3: 0x00049A14	Table 530, p. 680
PCI Express Power Management Register (p=1–3)	pcie_port_num1: 0x00041A18, pcie_port_num2: 0x00045A18, pcie_port_num3: 0x00049A18	Table 531, p. 681
PCI Express Flow Control Register (p=1–3)	pcie_port_num1: 0x00041A20, pcie_port_num2: 0x00045A20, pcie_port_num3: 0x00049A20	Table 532, p. 682
PCI Express Replay Timeout (TO) Timer Register (p=1–3)	pcie_port_num1: 0x00041A24, pcie_port_num2: 0x00045A24, pcie_port_num3: 0x00049A24	Table 533, p. 682
PCI Express Ack Latency Timer Register (p=1–3)	pcie_port_num1: 0x00041A28, pcie_port_num2: 0x00045A28, pcie_port_num3: 0x00049A28	Table 534, p. 682
PCI Express Secondary Bus Register (p=1–3)	pcie_port_num1: 0x00041A2C, pcie_port_num2: 0x00045A2C, pcie_port_num3: 0x00049A2C	Table 535, p. 683
PCI Express BAR Control (Ports 1-3)		
PCI Express BAR1 Control Register (p=1–3)	pcie_port_num1: 0x00041804, pcie_port_num2: 0x00045804, pcie_port_num3: 0x00049804	Table 536, p. 683
PCI Express BAR2 Control Register (p=1–3)	pcie_port_num1: 0x00041808, pcie_port_num2: 0x00045808, pcie_port_num3: 0x00049808	Table 537, p. 684
PCI Express Expansion ROM BAR Control Register (p=1–3)	pcie_port_num1: 0x0004180C, pcie_port_num2: 0x0004580C, pcie_port_num3: 0x0004980C	Table 538, p. 684
PCI Express Interrupt (Ports 1-3)		
PCI Express Interrupt Cause Register (p=1–3)	pcie_port_num1: 0x00041900, pcie_port_num2: 0x00045900, pcie_port_num3: 0x00049900	Table 539, p. 684
PCI Express Interrupt Mask Register (p=1–3)	pcie_port_num1: 0x00041910, pcie_port_num2: 0x00045910, pcie_port_num3: 0x00049910	Table 540, p. 687
PCI Express Debug (Ports 1-3)		
PCI Express Debug Control Register (p=1–3)	pcie_port_num1: 0x00041A60, pcie_port_num2: 0x00045A60, pcie_port_num3: 0x00049A60	Table 541, p. 688
PCI Express Debug MAC Status 1 Register (p=1–3)	pcie_port_num1: 0x00041A84, pcie_port_num2: 0x00045A84, pcie_port_num3: 0x00049A84	Table 542, p. 688
PCI Express TL Control Register (p=1–3)	pcie_port_num1: 0x00041AB0, pcie_port_num2: 0x00045AB0, pcie_port_num3: 0x00049AB0	Table 543, p. 689

Copyright © 2015 Marvell

A.8.1 PCI Express Configuration Cycles Generation (Ports 1-3)

Table 450: PCI Express Configuration Address Register (p=1–3)
Offset: pcie_port_num1: 0x000418F8, pcie_port_num2: 0x000458F8, pcie_port_num3: 0x000498F8
Offset Formula: 0x000418F8+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31	ConfigEn	RW 0x0	Configuration Enable Bit
30:28	Reserved	RSVD 0x0	Reserved
27:24	ExtRegNum	RW 0x0	Extended Register Number
23:16	BusNum	RW 0x0	Bus Number
15:11	DevNum	RW 0x0	Device Number
10:8	FunctNum	RW 0x0	Function Number
7:2	RegNum	RW 0x0	Register Number
1:0	Reserved	RSVD 0x0	Reserved

Table 451: PCI Express Configuration Data Register (p=1–3)
Offset: pcie_port_num1: 0x000418FC, pcie_port_num2: 0x000458FC, pcie_port_num3: 0x000498FC
Offset Formula: 0x000418FC+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	ConfigData	RW 0x0	Write access to this register generates a corresponding Configuration TLP on the PCI Express port or an access to the PCI Express port configuration header registers.

A.8.2 PCI Express Configuration Header (Ports 1-3)

Table 452: PCI Express Device and Vendor ID Register (p=1–3)
Offset: pcie_port_num1: 0x00040000, pcie_port_num2: 0x00044000, pcie_port_num3: 0x00048000
Offset Formula: 0x00040000+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	DevID	RO 0x7846	Device ID
15:0	VenID	RO 0x11AB	Vendor ID This field identifies Marvell as the Vendor of the device.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 453: PCI Express Command and Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040004, pcie_port_num2: 0x00044004, pcie_port_num3: 0x00048004
Offset Formula: 0x00040004+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31	DetParErr	RW1C 0x0	Detected Parity Error This bit is set when the device receives a poisoned TLP. **NOTE:** The bit is set regardless of the state of the <PErrEn> bit in this register. Write 1 to clear.
30	SSysErr	RW1C 0x0	Signalled System Error This bit is set when the device sends an ERR_FATAL or ERR_NONFATAL message. This bit is not set if the <SErrEn> field in this register is de-asserted. Write 1 to clear.
29	RMAbort	RW1C 0x0	Received Master Abort This bit is set when the device, as a requester (master), receives a completion with the status Unsupported Request. Write 1 to clear.
28	RTAbort	RW1C 0x0	Received Target Abort This bit is set when the device, as a requester (master), receives a completion with the status Completer Abort. Write 1 to clear.
27	STarAbort	RW1C 0x0	Signaled Target Abort This bit is set when the device, as a completer (target), completes a transaction as a Completer Abort. Write 1 to clear.
26:25	Reserved	RSVD 0x0	Does not apply to PCI Express. These bits are hardwired to 0.
24	MasDataPerr	RW1C 0x0	Master Data Parity Error This bit is set when the device has poisoned data detected as a requestor (master). Set when the <PErrEn> field (bit[6]) is set and either: Poisoned completion is received for the PCI Express port. or Poisoned TLP is transmitted to the PCI Express port. Write 1 to clear.
23:21	Reserved	RSVD 0x0	These bits are hardwired to 0.
20	CapList	RO 0x1	Capability List Support This bit indicates that the device configuration header includes capability list. This bit is hardwired to 1, since this is always supported in PCI Express.
19	IntStat	RO 0x0	Interrupt Status When set, this bit indicates that an interrupt message is pending internally in the device. 0 = Disable: No interrupt asserted. 1 = Enable: Interrupt asserted.
18:11	Reserved	RSVD 0x0	These bits are hardwired to 0.

Copyright © 2015 Marvell

Table 453: PCI Express Command and Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040004, pcie_port_num2: 0x00044004, pcie_port_num3: 0x00048004
Offset Formula: 0x00040004+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
10	IntDis	RW 0x0	Interrupt Disable This bit controls the ability of the device to generate interrupt emulation messages. When set, interrupt messages are not generated. 0 = Enable: Interrupt messages enabled. 1 = Disable: Interrupt messages disabled.
9	Reserved	RSVD 0x0	Does not apply to PCI Express. This bit is hardwired to 0.
8	SErrEn	RW 0x0	This bit controls the ability of the device to report fatal and non-fatal errors to the Root Complex. This bit affects both assertion of SSysErr status bit[30] in this register and uncorrectable error message generation. **NOTE:** PCI Express uncorrectable error messages are reported if enabled either through this field or through fields <NFErrRepEn> or <FErrRepEn> in the PCI Express Device Control Status Register. 0 = Disable: SSysErr assertion is disabled. 1 = Enable: SSysErr assertion is enabled. In addition uncorrectable error messages generation is enabled.
7	Reserved	RSVD 0x0	Does not apply to PCI Express. This bit is hardwired to 0.
6	PErrEn	RW 0x0	Parity Error Enable This bit controls the ability of the device to respond to poisoned data errors as a requestor (master) on the PCI Express port. Controls MasDataPerr status bit assertion in the PCMDSTT register. **NOTE:** The setting of this bit does not affect the DetectedPErr status bit. 0 = Disable: MasDataPerr assertion is disabled. 1 = Enable: MasDataPerr assertion is enabled.
5:3	Reserved	RSVD 0x0	Does not apply to PCI Express devices. These bits are hardwired to 0.
2	MasEn	RW 0x0	Master Enable This bit controls the ability of the device to act as a master on the PCI Express port. When set to 0, no memory or I/O read/write request packets are generated to PCI Express. **NOTE:** Messages and completions are transmitted to the PCI Express regardless of the setting of this bit. 0 = Disable: Neither memory nor I/O requests are transmitted to the PCI-E port. 1 = Enable: Memory and I/O requests are transmitted to the PCI-E port.
1	MemEn	RW 0x0	Memory Space Enable Controls the device response to PCI Express memory accesses. 0 = Disable: All Memory accesses from PCI Express are completed as Unsupported Requests. 1 = Enable
0	IOEn	RW 0x0	I/O Space Enable The I/O space is not enabled; therefore, this bit is not functional. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 454: PCI Express Class Code and Revision ID Register (p=1–3)
Offset: pcie_port_num1: 0x00040008, pcie_port_num2: 0x00044008, pcie_port_num3: 0x00048008
Offset Formula: 0x00040008+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:24	BaseClass	RO SAR	Device Base Class - Memory Controller
23:16	SubClass	RO SAR	Device Sub class--Other Memory Controller
15:8	ProgIF	RO 0x0	Register Level Programming Interface
7:0	RevID	RO 0x0	Device Revision Number

Table 455: PCI Express BIST Header Type and Cache Line Size Register (p=1–3)
Offset: pcie_port_num1: 0x0004000C, pcie_port_num2: 0x0004400C, pcie_port_num3: 0x0004800C
Offset Formula: 0x0004000C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31	BISTCap	RO 0x0	BIST Capable Bit BIST is not supported. **NOTE:** This bit must be set to 0.
30	BISTAct	RO 0x0	BIST Activate bit BIST is not supported.
29:28	Reserved	RSVD 0x0	Reserved
27:24	BISTComp	RO 0x0	BIST Completion Code 0x0 = BIST passed. Other = BIST failed.
23:16	HeadType	RO 0x0	Device Configuration Header Type The header is a Type 0 single-function configuration header.
15:8	Reserved	RSVD 0x0	Does not apply to PCI Express. These bits are hardwired to 0.
7:0	CacheLine	RW 0x00	Device Cache Line Size **NOTE:**

Table 456: PCI Express BAR0 Internal Register (p=1–3)
Offset: pcie_port_num1: 0x00040010, pcie_port_num2: 0x00044010, pcie_port_num3: 0x00048010
Offset Formula: 0x00040010+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:20	Base	RW 0xD00	Internal register memory Base Address Indicates 1-MByte address space. Corresponds to address bits[31:20].

Copyright © 2015 Marvell

Table 456: PCI Express BAR0 Internal Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040010, pcie_port_num2: 0x00044010, pcie_port_num3: 0x00048010
Offset Formula: 0x00040010+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
19:4	Reserved	RSVD 0x0	Reserved
3	Prefetch	RO 0x1	Prefetch Enable Indicates that prefetching is enabled.
2:1	Type	RO 0x2	BAR Type. BAR can be located in 64-bit memory address space.
0	Space	RO 0x0	Memory Space Indicator.

Table 457: PCI Express BAR0 Internal (High) Register (p=1–3)
Offset: pcie_port_num1: 0x00040014, pcie_port_num2: 0x00044014, pcie_port_num3: 0x00048014
Offset Formula: 0x00040014+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	Base	RW 0x0	Base address Corresponds to address bits[63:32].

Table 458: PCI Express BAR1 Register (p=1–3)
Offset: pcie_port_num1: 0x00040018, pcie_port_num2: 0x00044018, pcie_port_num3: 0x00048018
Offset Formula: 0x00040018+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base/Reserved	RW 0x0000	Base address Defined according to PCI Express BAR1 Control Register. Indicates 64-KByte to 4-GByte address space. Corresponds to address bits[31:16].
15:4	Reserved	RSVD 0x0	Reserved
3	Prefetch	RO 0x1	Prefetch Enable Indicates that prefetching is enabled.
2:1	Type	RO 0x2	BAR Type BAR can be located in 64-bit memory address space.
0	Space	RO 0x0	Memory Space Indicator

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 459: PCI Express BAR1 (High) Register (p=1–3)
Offset: pcie_port_num1: 0x0004001C, pcie_port_num2: 0x0004401C, pcie_port_num3: 0x0004801C
Offset Formula: 0x0004001C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	Base	RW 0x0	Base address Corresponds to address bits[63:32].

Table 460: PCI Express BAR2 Register (p=1–3)
Offset: pcie_port_num1: 0x00040020, pcie_port_num2: 0x00044020, pcie_port_num3: 0x00048020
Offset Formula: 0x00040020+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base/Reserved	RW 0xF000	Base address Defined according to PCI Express BAR2 Control Register. Indicates 64-KByte up to 4-GByte address space. Corresponds to address bits[31:16].
15:4	Reserved	RSVD 0x0	Reserved
3	Prefetch	RO 0x1	Prefetch Enable Indicates that pre-fetching is enabled.
2:1	Type	RO 0x2	BAR Type BAR can be located in 64-bit memory address space.
0	Space	RO 0x0	Memory Space Indicator

Table 461: PCI Express BAR2 (High) Register (p=1–3)
Offset: pcie_port_num1: 0x00040024, pcie_port_num2: 0x00044024, pcie_port_num3: 0x00048024
Offset Formula: 0x00040024+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	Base	RW 0x0	Base address Corresponds to address bits[63:32].

Table 462: PCI Express Subsystem Device and Vendor ID Register (p=1–3)
Offset: pcie_port_num1: 0x0004002C, pcie_port_num2: 0x0004402C, pcie_port_num3: 0x0004802C
Offset Formula: 0x0004002C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	SSDevID	RO 0x11AB	Subsystem Device ID Number The default is the Marvell Device ID.

Copyright © 2015 Marvell

Table 462: PCI Express Subsystem Device and Vendor ID Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x0004002C, pcie_port_num2: 0x0004402C, pcie_port_num3: 0x0004802C
Offset Formula: 0x0004002C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
15:0	SSVenID	RO 0x11AB	Subsystem Manufacturer ID Number The default is the Marvell Vendor ID.

Table 463: PCI Express Expansion ROM BAR Register (p=1–3)
Offset: pcie_port_num1: 0x00040030, pcie_port_num2: 0x00044030, pcie_port_num3: 0x00048030
Offset Formula: 0x00040030+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
If expansion ROM is not enabled via the ExpROMEn bit[0] the PCI Express Expansion ROM BAR Control Register, this register is reserved.			
31:22	RomBase	RW 0x0	Expansion ROM Base Address[31:22]
21:19	RomBase/Reserved	RW 0x0	These bits are defined according to field <ExpROMSz> in the PCI Express Expansion ROM BAR Control Register. When ROM Size is: 0 = 512 KByte Space: Bits[21:19] are used as Expansion ROM Base Address[21:19]. 1 = 1 MByte Space: Bits[21:20] are used as Expansion ROM Base Address[21:20], and bit[19] is reserved. 2 = 2 MByte Space: Bits[21] used as Expansion ROM Base Address[21], and bits[20:19] are reserved. 3 = 4 MByte Space: Bits[21:19] are reserved.
18:1	Reserved	RSVD 0x0	Reserved
0	RomEn	RW 0x0	Expansion ROM Enable **NOTE:** The device expansion ROM address space is enabled only if both this bit (RomEn) and <MemEn> in the PCI Express Command and Status Register are set. 0 = Disable 1 = Enable

Table 464: PCI Express Capability List Pointer Register (p=1–3)
Offset: pcie_port_num1: 0x00040034, pcie_port_num2: 0x00044034, pcie_port_num3: 0x00048034
Offset Formula: 0x00040034+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	CapPtr	RO 0x40	Capability List Pointer The current value in this field points to the PCI Power Management capability set in the PCI Express Power Management Capability Header Register at offset 0x40.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 465: PCI Express Interrupt Pin and Line Register (p=1–3)
Offset: pcie_port_num1: 0x0004003C, pcie_port_num2: 0x0004403C, pcie_port_num3: 0x0004803C
Offset Formula: 0x0004003C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Does not apply to PCI Express This field is hardwired to 0.
15:8	IntPin	RO 0x01	Indicates that the device is using INTA (Interrupt A) in the interrupt emulation messages.
7:0	IntLine	RW 0x00	Provides interrupt line routing information.

Table 466: PCI Express Power Management Capability Header Register (p=1–3)
Offset: pcie_port_num1: 0x00040040, pcie_port_num2: 0x00044040, pcie_port_num3: 0x00048040
Offset Formula: 0x00040040+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:27	PMESup	RO 0x00	This field indicates whether the device supports PM Event generation. The device does not support the PMEn pin.
26	D2Sup	RO 0x1	This bit indicates whether the device supports D2 Power Management state. The device does not support D2 state.
25	D1Sup	RO 0x1	This bit indicates whether the device supports D1 Power Management state. The device does not support D1 state.
24:22	AuxCur	RO 0x0	Auxiliary Current Requirements
21	DSI	RO 0x0	Device Specific Initialization The device does not requires device specific initialization.
20:19	Reserved	RSVD 0x0	Does not apply to PCI Express This field must be hardwired to 0.
18:16	PMCVer	RO 0x3	PCI Power Management Capability Version
15:8	NextPtr	RO 0x50	Next Item Pointer Current value points to MSI capability.
7:0	CapID	RO 0x01	Capability ID Current value identifies the PCI Power Management capability.

Copyright © 2015 Marvell

Table 467: PCI Express Power Management Control and Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040044, pcie_port_num2: 0x00044044, pcie_port_num3: 0x00048044
Offset Formula: 0x00040044+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:24	PMData	RO 0x0	State Data This field is used to report the state dependent data requested by the <PMDataSel> field of this register. The value of this field is scaled by the value reported by the <PMDataScale> field of this register.
23:16	Reserved	RSVD 0x0	Does not apply to PCI Express. This field must be hardwired to 0.
15	PMEStat	RW1C 0x0	PME Status **NOTE:** Sticky bit--not initialized by hot reset.
14:13	PMDataScale	RO 0x0	Data Scale Indicates the scaling factor to be used when interpreting the value of the <PMData> field of this register. The read value depends on the setting of the <PMDataSel> field of this register.
12:9	PMDataSel	RW 0x0	Data Select This 4-bit field is used to select which data is to be reported through the <PMDataScale> and <PMData> fields of this register.
8	PME_en	RW 0x0	PM_PME Message Generation Enable **NOTE:** Sticky bit--not initialized by hot reset. 0 = Disable 1 = Enable
7:4	Reserved	RSVD 0x0	Reserved
3	No_Soft_Reset	RO 0x0	When set (1), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. When clear (0), devices do perform an internal reset upon transitioning from D3hot to D0 via software control of the PowerState bits. Configuration Context is lost when performing the soft reset. Upon transition from the D3hot to the D0 state, full reinitialization sequence is needed to return the device to D0 Initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 uninitialized with only PME context preserved if PME is supported and enabled.
2	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 467: PCI Express Power Management Control and Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040044, pcie_port_num2: 0x00044044, pcie_port_num3: 0x00048044
Offset Formula: 0x00040044+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
1:0	PMState	RW 0x0	Power State This field controls the Power Management state of the device. The device supports all Power Management states. 0 = D0 1 = D1 2 = D2 3 = D3

Table 468: PCI Express MSI Message Control Register (p=1–3)
Offset: pcie_port_num1: 0x00040050, pcie_port_num2: 0x00044050, pcie_port_num3: 0x00048050
Offset Formula: 0x00040050+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23	Addr64	RO 0x1	64-bit Addressing Capable This field indicates whether the device is capable of generating a 64-bit message address. 0 = NotCapable 1 = Capable
22:20	MultiEn	RW 0x0	Multiple Messages Enable The number of messages the system allocates to the device. This number must be smaller or equal to the value that is in the <MultiCap> field.
19:17	MultiCap	RO 0x0	Multiple Message Capable The device is capable of driving a single message.
16	MSIEn	RW 0x0	MSI Enable Controls the device interrupt signaling mechanism. 0 = Disable: Interrupts are signaled through the interrupt emulation messages. 1 = Enable: Interrupts are signaled through MSI mechanism.
15:8	NextPtr	RO 0x60	Next Item Pointer The current value of this field points to PCI Express capability.
7:0	CapID	RO 0x5	Capability ID The current value of this field identifies the PCI MSI capability.

Table 469: PCI Express MSI Message Address Register (p=1–3)
Offset: pcie_port_num1: 0x00040054, pcie_port_num2: 0x00044054, pcie_port_num3: 0x00048054
Offset Formula: 0x00040054+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:2	MSIAddr	RW 0x0	Message Address This field corresponds to Address[31:2] of the MSI MWr TLP.

Copyright © 2015 Marvell

Table 469: PCI Express MSI Message Address Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040054, pcie_port_num2: 0x00044054, pcie_port_num3: 0x00048054
Offset Formula: 0x00040054+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
1:0	Reserved	RSVD 0x0	Reserved

Table 470: PCI Express MSI Message Address (High) Register (p=1–3)
Offset: pcie_port_num1: 0x00040058, pcie_port_num2: 0x00044058, pcie_port_num3: 0x00048058
Offset Formula: 0x00040058+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	MSIAddrh	RW 0x0	Message Upper Address This field corresponds to Address[63:32] of the MSI MWr TLP.

Table 471: PCI Express MSI Message Data Register (p=1–3)
Offset: pcie_port_num1: 0x0004005C, pcie_port_num2: 0x0004405C, pcie_port_num3: 0x0004805C
Offset Formula: 0x0004005C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	MSIData	RW 0x0	Message Data **NOTE:**

Table 472: PCI Express Capability Register (p=1–3)
Offset: pcie_port_num1: 0x00040060, pcie_port_num2: 0x00044060, pcie_port_num3: 0x00048060
Offset Formula: 0x00040060+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:25	IntMsgNum	RO 0x0	Interrupt Message Number
24	SlotImp	RO 0x0	Slot Implemented
23:20	DevType	RO SAR	Device/Port Type 1 = EndPoint: Legacy PCI Express Endpoint 4 = RootComplex: Root Port of PCI Express Root Complex

Copyright © 2015 Marvell

Table 472: PCI Express Capability Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040060, pcie_port_num2: 0x00044060, pcie_port_num3: 0x00048060
Offset Formula: 0x00040060+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
19:16	CapVer	RO 0x2	Capability Version This field indicates the PCI Express Base spec 2.0 version of the PCI Express capability.
15:8	NextPtr	RO 0x0	Next Item Pointer The current value of this field points to the end of the capability list (NULL).
7:0	CapID	RO 0x10	Capability ID The current value of this field identifies the PCI PE capability.

Table 473: PCI Express Device Capabilities Register (p=1–3)
Offset: pcie_port_num1: 0x00040064, pcie_port_num2: 0x00044064, pcie_port_num3: 0x00048064
Offset Formula: 0x00040064+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28	FuncLevelRstCpb	RO 0x0	Function Level Reset Capability. A value of 1b indicates the Function supports the optional Function Level Reset mechanism.
27:26	CapSPLScl	RO 0x0	Captured Slot Power Limit Scale
25:18	CapSPLVal	RO 0x0	Captured Slot Power Limit Value
17:16	Reserved	RSVD 0x0	Reserved
15	RuleBaseErrorRep	RO 0x1	Role-Based Error Reporting => This bit, when set, indicates that the device implements the functionality originally defined in the Error Reporting ECN for PCI Express Base Specification, Revision 1.0a, and later incorporated into PCI Express Base Specification, Revision 1.1. This bit must be set by all devices conforming to the ECN, PCI Express Base Specification, Revision 1.1., or subsequent PCI Express Base Specification revisions.
14	Reserved	RO 0x0	Reserved
13	Reserved	RO 0x0	Reserved
12	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 473: PCI Express Device Capabilities Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040064, pcie_port_num2: 0x00044064, pcie_port_num3: 0x00048064
Offset Formula: 0x00040064+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
11:9	EPL1AccLat	RO 0x0	Endpoint L1 Acceptable Latency This field indicates the amount of latency that can be absorbed by the device due to the transition from L1 to L0. The current value specifies less than 1 second. 000b Maximum of 1 µs 001b Maximum of 2 µs 010b Maximum of 4 µs 011b Maximum of 8 µs 100b Maximum of 16 µs 101b Maximum of 32 µs 110b Maximum of 64 µs 111b No limit
8:6	EPL0sAccLat	RO 0x2	Endpoint L0s Acceptable Latency This field indicates the amount of latency that can be absorbed by the device due to the transition from L0s to L0. 0 = Under64ns: Less than 64 ns. 1 = 64to128ns: Less than 128 ns. 2 = 128to256ns: Less than 256 ns. 3 = 256to512ns: Less than 512 ns. 4 = 512nsto1us: Less than 1 us. 5 = 1to2us: Less than 2 us. 6 = 2to4us: Less than 4 us. 7 = Above4us: No limit
5:3	Reserved	RO 0x0	Reserved These bits are hardwired to 0.
2:0	MaxPldSizeSup	RO 0x0	Maximum Payload Size Supported 128B MPS support. 0 = 128

Table 474: PCI Express Device Control Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040068, pcie_port_num2: 0x00044068, pcie_port_num3: 0x00048068
Offset Formula: 0x00040068+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved
21	TransPend	RO 0x0	Transactions Pending Indicates that the device has issued Non-Posted requests that have not been completed. 0 = Completed: All NP requests have been completed or terminated by the Completion Timeout Mechanism. 1 = Uncompleted: Not all NP requests have been completed or terminated.
20	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 474: PCI Express Device Control Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040068, pcie_port_num2: 0x00044068, pcie_port_num3: 0x00048068
Offset Formula: 0x00040068+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
19	URDet	RW1C 0x0	Unsupported Request Detected Indicates that the device received an unsupported request. Write 1 to clear.
18	FErrDet	RW1C 0x0	Fatal Error Detected Indicates the status of the fatal errors detected by the device. Write 1 to clear.
17	NFErrDet	RW1C 0x0	Non-Fatal Error Detected Indicates the status of the non-fatal errors detected by the device. Write 1 to clear.
16	CorErrDet	RW1C 0x0	Correctable Error Detected Indicates the status of the correctable errors detected by the device. Write 1 to clear.
15	Reserved	RO 0x0	Reserved
14:12	MaxRdRqSz	RW 0x2	Maximum Read Request Size Limits the device maximum read request size as a requestor (master). Other = Reserved 0 = 128B 1 = 256B 2 = 512B 3 = 1 KByte 4 = 2 KByte 5 = 4 KByte
11	EnNS	RO 0x0	Enable No Snoop The device never sets the No Snoop attribute in transactions it initiates as a requester. Hardwired to 0x0. 0 = Disable 1 = Enable
10	Reserved	RSVD 0x0	Reserved
9:8	Reserved	RO 0x0	Reserved These bits are hardwired to 0.
7:5	MaxPldSz	RW 0x0	Maximum Payload Size The maximum payload size supported is 128B (refer to bit <MaxPldSizeSup> in the PCI Express Device Capabilities Register). Other values = Reserved 0 = 128B: 0
4	EnRO	RO 0x0	Enable Relaxed Ordering The device never sets the Relaxed Ordering attribute in transactions it initiates as a requester. Hardwired to 0x0. 0 = Disable 1 = Enable

Table 474: PCI Express Device Control Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040068, pcie_port_num2: 0x00044068, pcie_port_num3: 0x00048068
Offset Formula: 0x00040068+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
3	URRepEn	RW 0x0	Unsupported Request (UR) Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated, therefore always write 0x0. **NOTE:** UR related error messages are still enabled when URRepEn=0, if <SErrEn> bit in PCI Express Command and Status Register is set. 0 = Disabled: UR related error messages are masked. Status bit is not masked. 1 = Enabled: UR related error messages enabled.
2	FErrRepEn	RW 0x0	Fatal Error Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated; therefore, always write 0x0. **NOTE:** ERR_FATAL error messages are still enabled when this field is 0, if the <SErrEn> bit in PCI Express Command and Status Register is set. 0 = Disable: ERR_FATAL error messages are masked. Status bit is still affected. 1 = Enable: ERR_FATAL error messages enabled.
1	NFErrRepEn	RW 0x0	Non-Fatal Error Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated, therefore always write 0x0. **NOTE:** ERR_NONFATAL error messages are still enabled when this field is 0, if the <SErrEn> bit in PCI Express Command and Status Register is set. 0 = Disable: ERR_NONFATAL error messages are masked. Status bit is not masked. 1 = Enable: ERR_NONFATAL error messages enabled.
0	CorErrRepEn	RW 0x0	Correctable Error Reporting Enable In Root Complex mode, reporting of errors is internal to status registers only. An external error message must not be generated; therefore, always write 0x0. **NOTE:** ERR_NONFATAL error messages are still enabled when this field is 0, if the <SErrEn> bit in the PCI Express Command and Status Register is set. 0 = Disable: ERR_COR error messages are masked. Status bit is not masked. 1 = Enable: ERR_COR error messages enabled.

Table 475: PCI Express Link Capabilities Register (p=1–3)
Offset: pcie_port_num1: 0x0004006C, pcie_port_num2: 0x0004406C, pcie_port_num3: 0x0004806C
Offset Formula: 0x0004006C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:24	PortNum	RO 0x0	In Endpoint mode, indicates the PCI Express port number as was advertised by the Root Complex during the link training process.

Document Classification: Proprietary Information

Table 475: PCI Express Link Capabilities Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x0004006C, pcie_port_num2: 0x0004406C, pcie_port_num3: 0x0004806C
Offset Formula: 0x0004006C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
23:19	Reserved	RSVD 0x0	Reserved
18	ClockPowerMng	RO 0x1	Clock Power Management A value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the clock request (CLKREQ#) mechanism when the link is in the L1 and L2/3 Ready link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these link states. This capability is applicable only in form factors that support clock request (CLKREQ#) capability. For a multi-function device, each function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the multifunction device indicate a 1b in this bit.
17:15	L1ExtLat	RO 0x7	L1 Exit Latency This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this port requires to complete transition from L1 to L0. Defined encodings are: 000b = Less than 1µs 001b = 1 µs to less than 2 µs 010b = 2 µs to less than 4 µs 011b = 4 µs to less than 8 µs 100b = 8 µs to less than 16 µs 101b = 16 µs to less than 32 µs 110b = 32 µs-64 µs 111b = More than 64 µs **NOTE:** The exit latencies can be influenced by the PCI Express reference clock configuration, depending upon whether a component uses a common or separate reference clock.
14:12	L0sExtLat	RO 0x2	L0s Exit Latency The time required by the device to transition its Rx lanes from L0s to L0. 0 = Under64ns: Less than 64 ns 1 = 64to128ns: 64 ns-128 ns 2 = 128to256ns: 128 ns-256 ns 3 = 256to512ns: 256 ns-512 ns 4 = 512nsto1µs: 512 ns-1 µs 5 = 1to2µs: 1 µs-2 µs 6 = 2to4µs: 2 us-4 us 7 = Reserved
11:10	AspmSup	RO 0x3	Active State Link PM Support The current value identifies L0s and L1 Supported. 1 = L0s_Entry_Supported 3 = L0s_and_L1_Supported
9:4	MaxLnkWdth	RO SAR	Maximum Link Width supported by the port. The legal values are 1 for X1 link and 4 for X4 link.

Copyright © 2015 Marvell

Table 475: PCI Express Link Capabilities Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x0004006C, pcie_port_num2: 0x0004406C, pcie_port_num3: 0x0004806C
Offset Formula: 0x0004006C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
3:0	MaxLinkSpd	RO SAR	Maximum Link Speed The current value identifies the 2.5 Gbps link. 1 = 2.5G Support: 2.5 GTb/s Link speed supported. 2 = 5G Support: 5.0 GT/s and 2.5 GT/s Link speeds supported.

Table 476: PCI Express Link Control Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040070, pcie_port_num2: 0x00044070, pcie_port_num3: 0x00048070
Offset Formula: 0x00040070+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31	PCIeLinkAutoBwStt	RW1C 0x0	Link Autonomous Bandwidth Status This bit is set by hardware to indicate that hardware has autonomously changed the link speed or width, without the port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable link operation. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was indicated as an autonomous change.
30	PCIeLinkBwMngStt	RW1C 0x0	Link Bandwidth Management Status – This bit is set by hardware to indicate that either of the following has occurred without the port transitioning through DL_Down status: - A Link retraining has completed following a write of 1b to the Retrain Link bit. **NOTE:** This bit is set following any write of 1b to the Retrain Link bit, including when the link is in the process of retraining for some other reason. - Hardware has changed the link speed or width to attempt to correct unreliable link operation, either through an LTSSM timeout or a higher level process. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change.
29	Reserved	RSVD 0x0	Reserved
28	SltClkCfg	RO 0x1	Slot Clock Configuration 0 = Independent: The device uses an independent clock, irrespective of the presence of a reference clock on the connector. 1 = Reference: The device uses the reference clock that the platform provides.
27	LnkTrn	RO 0x0	Link Training This bit is not applicable and reserved for Endpoints.
26	Reserved	RO 0x0	Reserved
25:20	NegLnkWdth	RO 0x0	Negotiated Link Width This field indicates the negotiated link width. This field is initialized by the hardware. The value in this field is undefined when the link is not up. 1 = X1 4 = X4

Copyright © 2015 Marvell

Table 476: PCI Express Link Control Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040070, pcie_port_num2: 0x00044070, pcie_port_num3: 0x00048070
Offset Formula: 0x00040070+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
19:16	LnkSpd	RO 0x1	Link Speed This field indicates the negotiated link speed. 1 = 2.5G: 2.5 GT/s PCI Express Link 2 = 5G: 5.0 GT/s PCI Express Link
15:10	Reserved	RSVD 0x0	Reserved
9	HW_autonomous_width_disable	RW 0x0	Hardware Autonomous Width Disable When set, this bit disables hardware from changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width.
8	EnClkPwrMng	RW 0x0	Enable Clock Power Management Applicable only for form factors that support a Clock Request (CLKREQ#) mechanism, this enable functions as follows: 0b = Clock power management is disabled and device must hold CLKREQ# signal low. 1b = When this bit is set to 1, the device is permitted to use CLKREQ# signal to power manage link clock according to protocol defined in appropriate form factor specification. Default value of this field is 0b. Components that do not support Clock Power Management (as indicated by a 0b value in the Clock Power Management bit of the Link Capabilities register) must hardwire this bit to 0b.
7	ExtdSnc	RW 0x0	Extended Sync When set, this bit forces extended transmission of 4096 FTS ordered sets followed by a single skip ordered set in exit from L0s and extra (1024) TS1 at exit from L1. **NOTE:** This bit is used for test and measurement.
6	CmnClkCfg	RW 0x0	Common Clock Configuration When set by the software, this bit indicates that both devices on the link use a distributed common reference clock.
5	RetrnLnk	RW 0x0	Retrain Link This bit forces the device to initiate link retraining. Always returns 0 when read. **NOTE:** If configured as an Endpoint, this field is reserved and has no effect.
4	LnkDis	RW 0x0	Link Disable **NOTE:** If configured as an Endpoint, this field is reserved and has no effect. Activation procedure: 1. Set this bit to trigger link disable. 2. Poll <DLDown> de-assertion (PCI Express Status Register, bit 0) ensure the link is disabled. 3. Clear the bit to exit to detect and enable the link again.

Table 476: PCI Express Link Control Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040070, pcie_port_num2: 0x00044070, pcie_port_num3: 0x00048070
Offset Formula: 0x00040070+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
3	RCB	RW 0x0	Read Completion Boundary **NOTE:** This bit has no effect on the device behavior. Completions are always returned with 128B read completion boundary. 0 = 64B: 64 Bytes 1 = 128B: 128 Bytes
2	Reserved	RSVD 0x0	Reserved
1:0	AspmCnt	RW 0x0	Active State Link PM Control Controls the level of active state PM supported on the link. 0 = Disable 1 = L0 Support: L0s entry supported. 2 = Reserved 3 = L0L1 Support: L0s and L1 entry supported.

Table 477: PCI Express Device Capabilities 2 Register (p=1–3)
Offset: pcie_port_num1: 0x00040084, pcie_port_num2: 0x00044084, pcie_port_num3: 0x00048084
Offset Formula: 0x00040084+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4	CplTODisSup	RO 0x0	Completion Timeout Disable Supported A value of 1b indicates support for the Completion Timeout Disable mechanism.
3:0	CplTORangesSup	RO 0x0	Completion Timeout Ranges Supported This field indicates device function support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. Four time value ranges are defined: Range A: 50 us to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4s Range D: 4 s to 64 s Bits are set according to the table below to show the timeout value ranges supported. 0000b = Completion Timeout programming not supported. The function must implement a timeout value in the range 50 us to 50 ms. 0001b = Range A 0010b = Range B 0011b = Ranges A and B 0110b = Ranges B and C 0111b = Ranges A, B, and C 1110b = Ranges B, C and D 1111b = Ranges A, B, C, and D All other values are reserved.

Copyright © 2015 Marvell

Table 478: PCI Express Device Control Status 2 Register (p=1–3)
Offset: pcie_port_num1: 0x00040088, pcie_port_num2: 0x00044088, pcie_port_num3: 0x00048088
Offset Formula: 0x00040088+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved
4	CplTODis	RW 0x0	Completion Timeout Disable When set, this bit disables the Completion Timeout mechanism. Software is permitted to set or clear this bit at any time. When set, the Completion Timeout detection mechanism is disabled. If there are outstanding requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding requests. If this is done, it is permitted to base the start time for each request on either the time this bit was cleared or the time each request was issued.
3:0	CplTOValue	RW 0x0	Completion Timeout Value Defined encoding: 0000b Default range: 50 us to 50 ms **NOTE:** Marvell strongly recommends that the Completion Timeout mechanism not expire in less than 10 ms. Values available if Range A (50 us to 10 ms) programmability range is supported: 0001b = 50 us to 100 us 0010b = 1 ms to 10 ms Values available if Range B (10 ms to 250 ms) programmability range is supported: 0101b = 16 ms to 55 ms 0110b = 65 ms to 210 ms Values available if Range C (250 ms to 4 s) programmability range is supported: 1001b = 260 ms to 900 ms 1010b = 1s to 3.5s Values available if the Range D (4 s to 64 s) programmability range is supported: 1101b = 4s to 13s 1110b = 17s to 64s Values not defined above are reserved.

Table 479: PCI Express Link Control Status 2 Register (p=1–3)
Offset: pcie_port_num1: 0x00040090, pcie_port_num2: 0x00044090, pcie_port_num3: 0x00048090
Offset Formula: 0x00040090+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved

Table 479: PCI Express Link Control Status 2 Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040090, pcie_port_num2: 0x00044090, pcie_port_num3: 0x00048090
Offset Formula: 0x00040090+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
16	CurrentDeemphasis Level	RO 0x0	Current De-emphasis Level When the Link operates at 5 GT/s speed, this bit reflects the level of de-emphasis. Values: 1b -3.5 dB 0b -6 dB The value in this bit is undefined when the link operates at 2.5GT/s speed.
15:13	Reserved	RSVD 0x0	Reserved
12	ComplianceDeemphasis	RW 0x0	Compliance De-emphasis This bit sets the de-emphasis level in Polling. Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Values: 1b -3.5 dB 0b -6 dB This bit is intended for debug, compliance testing purposes. System firmware and software are allowed to modify this bit only during debug or compliance testing.
11	ComplianceSOS	RW 0x0	Compliance SOS When set to 1b, the LTSSM is required to send SKP Ordered Sets periodically in between the (modified) compliance patterns.
10	EnterModifiedCompliance	RW 0x0	Enter Modified Compliance When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. This fiel is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value.
9:7	TransmitMargin	RO 0x0	Transmit Margin This field controls the value of the nonde-emphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see the relevant section in the Functional Specifications about how the transmitter voltage level is determined in various states). Values: a) 000b = Normal operating range b) 001b = 800-1200 mV for full swing and 400-700 mV for half-swing c) 010b - (n-1) = Values must be monotonic with a non-zero slope. The value of n must be greater than 3 and less than 7. At least two of these must be below the normal operating range of n: 200-400 mV for full-swing and 100-200 mV for half-swing d) (n) - 111b = Reserved **NOTE:** This register can be written to, although it is appears as RO. This register is intended for debug, compliance testing purposes only. System firmware and software are allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.

Copyright © 2015 Marvell

Table 479: PCI Express Link Control Status 2 Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040090, pcie_port_num2: 0x00044090, pcie_port_num3: 0x00048090
Offset Formula: 0x00040090+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
6	SelectDeemphasis	RO 0x0	Selectable De-emphasis When the link is operating at 5.0 GT/s speed, this bit selects the level of de-emphasis for an Upstream component. Values: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream ports of switches. This field can be changed from the Mbus.
5	HardwareAutoSpeedDis	RO 0x0	Hardware Autonomous Speed Disable When set, this bit disables hardware from changing the link speed for device specific reasons other than attempting to correct unreliable link operation by reducing link speed. Initial transition to the highest supported common link speed is not blocked by this bit. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. PCIe does not implement this feature.
4	EnterCompliance	RW 0x0	Enter Compliance Software is permitted to force a link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link.
3:0	TargetLinkSpeed	RW 0x2	Target Link Speed - For Downstream Ports, this field sets an upper limit on link operational speed by restricting the values advertised by the Upstream component in its training sequences. Defined values: 0001b = 2.5 GT/s Target Link Speed 0010b = 5.0 GT/s Target Link Speed All other values are reserved. If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, the result is undefined. The default value of this field is the highest link speed supported by the component (as reported in the Supported Link Speeds field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value. For both Upstream and Downstream ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode.

Copyright © 2015 Marvell

Table 480: PCI Express Advanced Error Report Header Register (p=1–3)
Offset: pcie_port_num1: 0x00040100, pcie_port_num2: 0x00044100, pcie_port_num3: 0x00048100
Offset Formula: 0x00040100+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:20	NextPtr	RO 0x0	Next Item Pointer This field indicates the last item in the extended capabilities linked list.
19:16	CapVer	RO 0x1	Capability Version
15:0	PECapID	RO 0x1	Extended Capability ID The current value of this field identifies the Advanced Error Reporting capability.

Table 481: PCI Express Uncorrectable Error Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040104, pcie_port_num2: 0x00044104, pcie_port_num3: 0x00048104
Offset Formula: 0x00040104+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset. All fields in this register (except for reserved fields) are SC. Write 1 to clear. A write of 0 has no effect.			
31:21	Reserved	RSVD 0x0	Reserved
20	UREr	RW1C 0x0	Unsupported Request Error Status **NOTE:** Sticky bit--not initialized by hot reset.
19	Reserved	RSVD 0x0	Reserved
18	MalfTlpErr	RW1C 0x0	Malformed TLP Status **NOTE:** Sticky bit--not initialized by hot reset.
17	Reserved	RW1C 0x0	Reserved **NOTE:** Sticky bit--not initialized by hot reset. **NOTE:**
16	UnexpCmpErr	RW1C 0x0	Unexpected Completion Status **NOTE:** Sticky bit--not initialized by hot reset.
15	CAErr	RW1C 0x0	Completer Abort Status **NOTE:** Sticky bit--not initialized by hot reset.
14	CmpTOErr	RW1C 0x0	Completion Timeout Status **NOTE:** Sticky bit--not initialized by hot reset.
13	Reserved	RSVD 0x0	Reserved **NOTE:** Sticky bit--not initialized by hot reset. **NOTE:**
12	RPsnTlpErr	RW1C 0x0	Poisoned TLP Status **NOTE:** Hot sticky bit--not initialized by hot reset.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 481: PCI Express Uncorrectable Error Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040104, pcie_port_num2: 0x00044104, pcie_port_num3: 0x00048104
Offset Formula: 0x00040104+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
11:5	Reserved	RSVD 0x0	Reserved
4	DLPrtErr	RW1C 0x0	Data Link Protocol Error Status **NOTE:** Hot sticky bit--not initialized by hot reset.
3:0	Reserved	RSVD 0x0	Reserved

Table 482: PCI Express Uncorrectable Error Mask Register (p=1–3)
Offset: pcie_port_num1: 0x00040108, pcie_port_num2: 0x00044108, pcie_port_num3: 0x00048108
Offset Formula: 0x00040108+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky not initialized or modified by hot reset.			
31:21	Reserved	RSVD 0x0	Reserved
20	URErrMsk	RW 0x0	Unsupported Request Error Mask
19	Reserved	RSVD 0x0	Reserved.
18	MalfTlpErrMsk	RW 0x0	Malformed TLP Mask
17	Reserved	RW 0x0	Reserved
16	UnexpCmpErrMsk	RW 0x0	Unexpected Completion Mask
15	CAErrMsk	RW 0x0	Completer Abort Mask
14	CmpTOErrMsk	RW 0x0	Completion Timeout Mask
13	Reserved	RSVD 0x0	Reserved
12	RPsnTlpErrMsk	RW 0x0	Poisoned TLP Error Mask
11:5	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 482: PCI Express Uncorrectable Error Mask Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040108, pcie_port_num2: 0x00044108, pcie_port_num3: 0x00048108
Offset Formula: 0x00040108+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
4	DLPrtErrMsk	RW 0x0	Data Link Protocol Error Mask When an error is indicated in the PCI Express Uncorrectable Error Status Register and the corresponding bit is set: - The header is not logged in the Header Log Register. - The First Error Pointer is not updated. - An error message is not generated. Status bit is set regardless of the mask setting. 0 = NotMasked 1 = Masked
3:0	Reserved	RSVD 0x0	Reserved

Table 483: PCI Express Uncorrectable Error Severity Register (p=1–3)
Offset: pcie_port_num1: 0x0004010C, pcie_port_num2: 0x0004410C, pcie_port_num3: 0x0004810C
Offset Formula: 0x0004010C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are hot sticky not initialized or modified by reset.			
31:21	Reserved	RSVD 0x0	Reserved
20	URErrSev	RW 0x0	Unsupported Request Error Severity
19	Reserved	RSVD 0x0	Reserved
18	MalfTlpErrSev	RW 0x1	Malformed TLP Severity
17	Reserved	RW 0x1	Reserved
16	UnexpCmpErrSev	RW 0x0	Unexpected Completion Severity
15	CAErrSev	RW 0x0	Completer Abort Severity
14	CmpTOErrSev	RW 0x0	Completion Timeout Severity
13	Reserved	RSVD 0x0	Reserved
12	RPsnTlpErrSev	RW 0x0	Poisoned TLP Error Severity
11:5	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 483: PCI Express Uncorrectable Error Severity Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x0004010C, pcie_port_num2: 0x0004410C, pcie_port_num3: 0x0004810C
Offset Formula: 0x0004010C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
4	DLPrtErrSev	RW 0x1	Data Link Protocol Error Severity Controls the severity indication of the Uncorrectable errors. Each bit controls the error type of the corresponding bit in the PCI Express Uncorrectable Error Status Register. 0 = NonFatal: Error type is Non-Fatal. 1 = Fatal: Error type is Fatal.
3:0	Reserved	RSVD 0x0	Reserved

Table 484: PCI Express Correctable Error Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040110, pcie_port_num2: 0x00044110, pcie_port_num3: 0x00048110
Offset Formula: 0x00040110+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky not initialized or modified by hot reset. All fields in this register (except for reserved fields) are RW1C write 1 to clear.			
31:14	Reserved	RSVD 0x0	Reserved
13	AdvNonFatalErr	RW1C 0x0	Advisory Non-Fatal Error Status
12	RplyTOErr	RW1C 0x0	Replay Timer Timeout Status
11:9	Reserved	RSVD 0x0	Reserved
8	RplyRllovrErr	RW1C 0x0	Replay Number Rollover Status
7	BadDllpErr	RW1C 0x0	Bad DLLP Status
6	BadTlpErr	RW1C 0x0	Bad TLP Status
5:1	Reserved	RSVD 0x0	Reserved
0	RcvErr	RW1C 0x0	Receiver Error Status **NOTE:** When set, this bit indicates that a Receiver error has occurred.

Copyright © 2015 Marvell

Table 485: PCI Express Correctable Error Mask Register (p=1–3)
Offset: pcie_port_num1: 0x00040114, pcie_port_num2: 0x00044114, pcie_port_num3: 0x00048114
Offset Formula: 0x00040114+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky not initialized or modified by hot reset.			
31:14	Reserved	RSVD 0x0	Reserved
13	AdvNonFatalMsk	RW 0x1	Advisory Non-Fatal Error Mask This bit is set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.
12	RplyTOMsk	RW 0x0	Replay Timer Timeout Mask
11:9	Reserved	RSVD 0x0	Reserved
8	RplyRllovrMsk	RW 0x0	Replay Number Rollover Mask
7	BadDllpErrMsk	RW 0x0	Bad DLLP Mask
6	BadTlpMsk	RW 0x0	Bad TLP Mask
5:1	Reserved	RSVD 0x0	Reserved
0	RcvMsk	RW 0x0	Receiver Error Mask If set, an error message is not generated upon occurrence of a Receiver error. 0 = NotMasked 1 = Masked

Table 486: PCI Express Advanced Error Capability and Control Register (p=1–3)
Offset: pcie_port_num1: 0x00040118, pcie_port_num2: 0x00044118, pcie_port_num3: 0x00048118
Offset Formula: 0x00040118+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved **NOTE:** Bits [5] and [7] are Read Only and are hardwired to 0.

Copyright © 2015 Marvell

Table 486: PCI Express Advanced Error Capability and Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00040118, pcie_port_num2: 0x00044118, pcie_port_num3: 0x00048118
Offset Formula: 0x00040118+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
4:0	FrstErrPtr	RO 0x0	First Error Pointer This field reports the bit position of the first error reported in the PCI Express Uncorrectable Error Status Register. This field locks upon receipt of the first uncorrectable error that is not masked. It remains locked until software clears it by writing 1 to the corresponding status bit. Upon receipt of the next uncorrectable error that is not masked, the field locks again until cleared as described above. This lock and clear process continues to repeat itself. **NOTE:** The bits in this field are sticky bits--they are not initialized or modified by reset. 4 = DLP: Data Link Protocol Error 12 = TLP: Poisoned TLP Error 13 = FCP: Flow Control Protocol Error 14 = CT: Completion Timeout Error 15 = CAS: Completer Abort Status 16 = UCE: Unexpected Completion Error 17 = ROE: Receiver Overflow Error 18 = Mutant TLP: Malformed TLP Error 19 = Reserved 20 = URE: Unsupported Request Error Other = Reserved

Table 487: PCI Express Header Log First DWORD Register (p=1–3)
Offset: pcie_port_num1: 0x0004011C, pcie_port_num2: 0x0004411C, pcie_port_num3: 0x0004811C
Offset Formula: 0x0004011C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog1DW	RO 0x0	Header Log First DWORD Logs the header of the first error reported in the PCI Express Uncorrectable Error Status Register (PEUEST). This field locks upon receipt of the first uncorrectable error that is not masked. It remains locked until software clears it by writing 1 to the corresponding status bit. Upon receipt of the next uncorrectable error that is not masked, the field locks again until cleared as described above. This lock and clear process continues to repeat itself. **NOTE:**

Table 488: PCI Express Header Log Second DWORD Register (p=1–3)
Offset: pcie_port_num1: 0x00040120, pcie_port_num2: 0x00044120, pcie_port_num3: 0x00048120
Offset Formula: 0x00040120+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog2DW	RO 0x0	Header Log Second DWORD **NOTE:**

Copyright © 2015 Marvell

Table 489: PCI Express Header Log Third DWORD Register (p=1–3)
Offset: pcie_port_num1: 0x00040124, pcie_port_num2: 0x00044124, pcie_port_num3: 0x00048124
Offset Formula: 0x00040124+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog3DW	RO 0x0	Header Log Third DWORD **NOTE:**

Table 490: PCI Express Header Log Fourth DWORD Register (p=1–3)
Offset: pcie_port_num1: 0x00040128, pcie_port_num2: 0x00044128, pcie_port_num3: 0x00048128
Offset Formula: 0x00040128+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All fields in this register are sticky, not initialized or modified by hot reset.			
31:0	Hdrlog4DW	RO 0x0	Header Log Fourth DWORD **NOTE:** All fields in this register are Hot sticky bit -- not initialized or modified by hot reset.

Table 491: PCI Express Root Error Command Register (p=1–3)
Offset: pcie_port_num1: 0x0004012C, pcie_port_num2: 0x0004412C, pcie_port_num3: 0x0004812C
Offset Formula: 0x0004012C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:3	Reserved	RO 0x0	Reserved
2	Fatal Error Reporting Enable	RW 0x0	Fatal Error Reporting Enable When set, this bit enables the generation of an interrupt when a fatal error is reported by any of the functions in the hierarchy associated with this Root Port.
1	Non-Fatal Error Reporting Enable	RW 0x0	Non-Fatal Error Reporting Enable When set, this bit enables the generation of an interrupt when a non-fatal error is reported by any of the functions in the hierarchy associated with this Root Port.
0	Correctable Error Reporting Enable	RW 0x0	Correctable Error Reporting Enable When set, this bit enables the generation of an interrupt when a correctable error is reported by any of the functions in the hierarchy associated with this Root Port.

Copyright © 2015 Marvell

Table 492: PCI Express Root Error Status Register (p=1–3)
Offset: pcie_port_num1: 0x00040130, pcie_port_num2: 0x00044130, pcie_port_num3: 0x00048130
Offset Formula: 0x00040130+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:27	Advanced Error Interrupt Message Number	RO 0x0	Advanced Error Interrupt Message Number This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this capability.
26:7	Reserved	RO 0x0	RsvdZ
6	Fatal Error Messages Received	RW1C 0x0	Fatal Error Messages Received Set when one or more fatal uncorrectable error messages have been received.
5	Non-Fatal Error Messages Received	RW1C 0x0	Non-Fatal Error Messages Received Set when one or more non-fatal uncorrectable error messages have been received.
4	First Uncorrectable Fatal	RW1C 0x0	First Uncorrectable Fatal Set when the first uncorrectable error message received is for a fatal error.
3	Multiple ERR_FATAL/NONFATAL Received	RW1C 0x0	Multiple ERR_FATAL/NONFATAL Received Set when either a fatal or a non-fatal error is received and ERR_FATAL/NONFATAL Received is already set.
2	ERR_FATAL/NONFATAL Received	RW1C 0x0	ERR_FATAL/NONFATAL Received Set when either a fatal or a non-fatal error message is received and this bit is not already set.
1	Multiple ERR_COR Received	RW1C 0x0	Multiple ERR_COR Received Set when a correctable error message is received and ERR_COR Received is already set.
0	ERR_COR Received	RW1C 0x0	ERR_COR Received Set when a correctable error message is received and this bit is not already set.

Table 493: PCI Express Error Source Identification Register (p=1–3)
Offset: pcie_port_num1: 0x00040134, pcie_port_num2: 0x00044134, pcie_port_num3: 0x00048134
Offset Formula: 0x00040134+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	ERR_FATAL/NONFATAL Source Identification	ROC 0x0	ERR_FATAL/NONFATAL Source Identification Loaded with the Requester ID indicated in the received ERR_FATAL or ERR_NONFATAL message when the ERR_FATAL/NONFATAL Received bit is not already set.
15:0	ERR_COR Source Identification	ROC 0x0	ERR_COR Source Identification Loaded with the Requester ID indicated in the received ERR_COR Message when the ERR_COR Received bit is not already set.

Copyright © 2015 Marvell

A.8.3 PCI Express Address Window Control (Ports 1-3)

Table 494: PCI Express Window0 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041820, pcie_port_num2: 0x00045820, pcie_port_num3: 0x00049820
Offset Formula: 0x00041820+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by a sequence of 0s. The number of 1s specifies the size of the window in 64-KByte granularity. (A value of 0x0FFF specifies 256 MByte).
15:8	Attr	RW 0x0E	Target specific attributes depending on the target interface. See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See the "PCI Express Address Decoding" section.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping to BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Table 495: PCI Express Window0 Base Register (p=1–3)
Offset: pcie_port_num1: 0x00041824, pcie_port_num2: 0x00045824, pcie_port_num3: 0x00049824
Offset Formula: 0x00041824+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0000	Base Address Used with the <Size> field to set the address window size and location. Corresponds to transaction address [31:16].
15:0	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 496: PCI Express Window0 Remap Register (p=1–3)
Offset: pcie_port_num1: 0x0004182C, pcie_port_num2: 0x0004582C, pcie_port_num3: 0x0004982C
Offset Formula: 0x0004182C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address Used with the <Size> field to specifies address bits[31:0] to be driven to the target interface.
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 497: PCI Express Window1 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041830, pcie_port_num2: 0x00045830, pcie_port_num3: 0x00049830
Offset Formula: 0x00041830+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size (A value of 0x0FFF specifies 256 MByte).
15:8	Attr	RW 0x0D	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 498: PCI Express Window1 Base Register (p=1–3)
Offset: pcie_port_num1: 0x00041834, pcie_port_num2: 0x00045834, pcie_port_num3: 0x00049834
Offset Formula: 0x00041834+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x1000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 499: PCI Express Window1 Remap Register (p=1–3)
Offset: pcie_port_num1: 0x0004183C, pcie_port_num2: 0x0004583C, pcie_port_num3: 0x0004983C
Offset Formula: 0x0004183C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 500: PCI Express Window2 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041840, pcie_port_num2: 0x00045840, pcie_port_num3: 0x00049840
Offset Formula: 0x00041840+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size
15:8	Attr	RW 0x0B	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 500: PCI Express Window2 Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041840, pcie_port_num2: 0x00045840, pcie_port_num3: 0x00049840
Offset Formula: 0x00041840+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
0	WinEn	RW 0x1	Window Enable 0 = Disable 1 = Enable

Table 501: PCI Express Window2 Base Register (p=1–3)
Offset: pcie_port_num1: 0x00041844, pcie_port_num2: 0x00045844, pcie_port_num3: 0x00049844
Offset Formula: 0x00041844+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x2000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 502: PCI Express Window2 Remap Register (p=1–3)
Offset: pcie_port_num1: 0x0004184C, pcie_port_num2: 0x0004584C, pcie_port_num3: 0x0004984C
Offset Formula: 0x0004184C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 503: PCI Express Window3 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041850, pcie_port_num2: 0x00045850, pcie_port_num3: 0x00049850
Offset Formula: 0x00041850+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size
15:8	Attr	RW 0x07	Target specific attributes depending on the target interface.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 503: PCI Express Window3 Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041850, pcie_port_num2: 0x00045850, pcie_port_num3: 0x00049850
Offset Formula: 0x00041850+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = 128B: Write split on 128-Byte address boundary. 1 = 32B: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable. Window is disabled by default. 0 = Disable 1 = Enable

Table 504: PCI Express Window3 Base Register (p=1–3)
Offset: pcie_port_num1: 0x00041854, pcie_port_num2: 0x00045854, pcie_port_num3: 0x00049854
Offset Formula: 0x00041854+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x3000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 505: PCI Express Window3 Remap Register (p=1–3)
Offset: pcie_port_num1: 0x0004185C, pcie_port_num2: 0x0004585C, pcie_port_num3: 0x0004985C
Offset Formula: 0x0004185C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Copyright © 2015 Marvell

Table 506: PCI Express Window4 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041860, pcie_port_num2: 0x00045860, pcie_port_num3: 0x00049860
Offset Formula: 0x00041860+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	Window Size (A value of 0x07FF specifies 128 MByte).
15:8	Attr	RW 0x3B	Target specific attributes depending on the target interface.
7:4	Target	RW 0x1	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = 128B: Write split on 128-Byte address boundary. 1 = 32B: Write split on 32-Byte address boundary.
1	BarMap	RW 0x1	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable 0 = Disable 1 = Enable

Table 507: PCI Express Window4 Base Register (p=1–3)
Offset: pcie_port_num1: 0x00041864, pcie_port_num2: 0x00045864, pcie_port_num3: 0x00049864
Offset Formula: 0x00041864+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xF000	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 508: PCI Express Window4 Remap Register (p=1–3)
Offset: pcie_port_num1: 0x0004186C, pcie_port_num2: 0x0004586C, pcie_port_num3: 0x0004986C
Offset Formula: 0x0004186C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 508: PCI Express Window4 Remap Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x0004186C, pcie_port_num2: 0x0004586C, pcie_port_num3: 0x0004986C
Offset Formula: 0x0004186C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 509: PCI Express Window4 Remap (High) Register (p=1–3)
Offset: pcie_port_num1: 0x00041870, pcie_port_num2: 0x00045870, pcie_port_num3: 0x00049870
Offset Formula: 0x00041870+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap High Address Specifies address bits[63:32] to be driven to the target interface. Only relevant for target interfaces that support a more than 4-GByte address space. When using target interface that do not support a more than 4-GByte address space, this register must be cleared.

Table 510: PCI Express Window5 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041880, pcie_port_num2: 0x00045880, pcie_port_num3: 0x00049880
Offset Formula: 0x00041880+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x07FF	Window Size (A value of 0x07FF specifies 128 MByte).
15:8	Attr	RW 0x2F	Target specific attributes depending on the target interface.
7:4	Target	RW 0x1	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = 128B: Write split on 128-Byte address boundary. 1 = 32B: Write split on 32-Byte address boundary.
1	BarMap	RW 0x1	Mapping To BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x1	Window Enable 0 = Disable 1 = Enable

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 511: PCI Express Window5 Base Register (p=1–3)
Offset: pcie_port_num1: 0x00041884, pcie_port_num2: 0x00045884, pcie_port_num3: 0x00049884
Offset Formula: 0x00041884+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0xF800	Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 512: PCI Express Window5 Remap Register (p=1–3)
Offset: pcie_port_num1: 0x0004188C, pcie_port_num2: 0x0004588C, pcie_port_num3: 0x0004988C
Offset Formula: 0x0004188C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 513: PCI Express Window5 Remap (High) Register (p=1–3)
Offset: pcie_port_num1: 0x00041890, pcie_port_num2: 0x00045890, pcie_port_num3: 0x00049890
Offset Formula: 0x00041890+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap High Address Specifies address bits[63:32] to be driven to the target interface. Only relevant for target interfaces that support a more than 4-GByte address space. When using target interface that do not support a more than 4-GByte address space, this register must be cleared.

Table 514: PCI Express Default Window Control Register (p=1–3)
Offset: pcie_port_num1: 0x000418B0, pcie_port_num2: 0x000458B0, pcie_port_num3: 0x000498B0
Offset Formula: 0x000418B0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	Attr	RW 0x0	Target specific attributes depending on the target interface.

Copyright © 2015 Marvell

Table 514: PCI Express Default Window Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x000418B0, pcie_port_num2: 0x000458B0, pcie_port_num3: 0x000498B0
Offset Formula: 0x000418B0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control Controls the address boundary on which slave write transactions are split, when forwarded to the internal interface. Only relevant for transactions that hit the respective address window. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1:0	Reserved	RSVD 0x0	Reserved

Table 515: PCI Express Expansion ROM Window Control Register (p=1–3)
Offset: pcie_port_num1: 0x000418C0, pcie_port_num2: 0x000458C0, pcie_port_num3: 0x000498C0
Offset Formula: 0x000418C0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	Attr	RW 0x2F	Target specific attributes depending on the target interface. Default corresponds to DEV_BOOTCSn.
7:4	Target	RW 0x1	Specifies the unit ID (target interface) associated with this window. Default represents the Device Bus controller.
3:0	Reserved	RSVD 0x0	Reserved

Table 516: PCI Express Expansion ROM Window Remap Register (p=1–3)
Offset: pcie_port_num1: 0x000418C4, pcie_port_num2: 0x000458C4, pcie_port_num3: 0x000498C4
Offset Formula: 0x000418C4+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Document Classification: Proprietary Information

Table 517: PCI Express Window<n+6> Control Register (n=0–9, p=1–3)
Offset: pcie_port_num1x10: 0x00041C00,
pcie_port_num1x11: 0x00041C10...pcie_port_num3x19: 0x00049C90
Offset Formula: 0x00041C00+0x10*n + 0x4000*(p-1) : where p (1-3) represents pcie_port_num, where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by a sequence of 0s. The number of 1s specifies the size of the window in 64-KByte granularity. (A value of 0x0FFF specifies 256 MByte).
15:8	Attr	RW 0x0E	Target specific attributes depending on the target interface. See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See the "PCI Express Address Decoding" section.
3	Reserved	RSVD 0x0	Reserved
2	SlvWrSpltCnt	RW 0x0	Slave Write Split Control When forwarded to the internal interface, controls the address boundary on which slave write transactions are split. Only relevant for transactions that hit the respective address window. **NOTE:** When working with cache coherency, this bit must be set to 1. 0 = SplitTo128: Write split on 128-Byte address boundary. 1 = SplitTo32: Write split on 32-Byte address boundary.
1	BarMap	RW 0x0	Mapping to BAR 0 = BAR1: Window is mapped to BAR1. 1 = BAR2: Window is mapped to BAR2.
0	WinEn	RW 0x0	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Table 518: PCI Express Window<n+6> Base Register (n=0–9, p=1–3)
Offset: pcie_port_num1x10: 0x00041C04,
pcie_port_num1x11: 0x00041C14...pcie_port_num3x19: 0x00049C94
Offset Formula: 0x00041C04+n*0x10 + 0x4000*(p-1) : where p (1-3) represents pcie_port_num, where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0000	Base Address Used with the <Size> field to set the address window size and location. Corresponds to transaction address [31:16].
15:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 519: PCI Express Window<n+6> Remap Register (n=0–9, p=1–3)
Offset: pcie_port_num1x10: 0x00041C08, pcie_port_num1x11: 0x00041C18...pcie_port_num3x19: 0x00049C98
Offset Formula: 0x00041C08+0x10*n + 0x4000*(p-1) : where p (1-3) represents pcie_port_num, where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:16	Remap	RW 0x0	Remap Address
15:1	Reserved	RSVD 0x0	Reserved
0	RemapEn	RW 0x0	Remap Enable Bit 0 = Disable: Remap disabled. 1 = Enable: Remap enabled.

Table 520: PCI Express Window<n+6> Remap (High) Register (n=0–9, p=1–3)
Offset: pcie_port_num1x10: 0x00041C0C, pcie_port_num1x11: 0x00041C1C...pcie_port_num3x19: 0x00049C9C
Offset Formula: 0x00041C0C+0x10*n + 0x4000*(p-1) : where p (1-3) represents pcie_port_num, where n (0-9) represents x1

Bit	Field	Type/InitVal	Description
31:0	RemapHigh	RW 0x0	Remap High Address Specifies address bits[63:32] to be driven to the target interface. Only relevant for target interfaces that support a more than 4-GByte address space. When using target interface that do not support a more than 4-GByte address space, this register must be cleared.

A.8.4 PCI Express Mbus Control (Ports 1-3)

Table 521: PCI Express Mbus Adapter Control Register (p=1–3)
Offset: pcie_port_num1: 0x000418D0, pcie_port_num2: 0x000458D0, pcie_port_num3: 0x000498D0
Offset Formula: 0x000418D0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:21	Reserved	RSVD 0x0	Reserved
20	TxDPPropEn	RW 0x1	Tx Data Poisoning Error Propagation Enable 0 = Enable 1 = Disable
19	RxDPPropEn	RW 0x1	Rx Data Poisoning Error Propagation Enable 0 = Disable 1 = Enable
18:6	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 521: PCI Express Mbus Adapter Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x000418D0, pcie_port_num2: 0x000458D0, pcie_port_num3: 0x000498D0
Offset Formula: 0x000418D0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
5	Reserved	RW 0x0	Reserved
4	Reserved	RW 0x0	Reserved
3:2	Reserved	RW 0x0	Reserved
1	Reserved	RW 0x0	Reserved
0	Reserved	RW 0x0	Reserved

Table 522: PCI Express Mbus Arbiter Control Register (Low) (p=1–3)
Offset: pcie_port_num1: 0x000418E0, pcie_port_num2: 0x000458E0, pcie_port_num3: 0x000498E0
Offset Formula: 0x000418E0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:28	Arb7	RW 0x7	Slice 7 of arbiter.
27:24	Arb6	RW 0x6	Slice 6 of arbiter.
23:20	Arb5	RW 0x5	Slice 5 of arbiter.
19:16	Arb4	RW 0x4	Slice 4 of arbiter.
15:12	Arb3	RW 0x3	Slice 3 of arbiter.
11:8	Arb2	RW 0x2	Slice 2 of arbiter.
7:4	Arb1	RW 0x1	Slice 1 of arbiter.
3:0	Arb0	RW 0x0	Slice 0 of arbiter.

Table 523: PCI Express Mbus Arbiter Control Register (High) (p=1–3)
Offset: pcie_port_num1: 0x000418E4, pcie_port_num2: 0x000458E4, pcie_port_num3: 0x000498E4
Offset Formula: 0x000418E4+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:28	Arb15	RW 0xf	Slice 15 of arbiter.

Copyright © 2015 Marvell

Table 523: PCI Express Mbus Arbiter Control Register (High) (p=1–3) (Continued)
Offset: pcie_port_num1: 0x000418E4, pcie_port_num2: 0x000458E4, pcie_port_num3: 0x000498E4
Offset Formula: 0x000418E4+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
27:24	Arb14	RW 0xe	Slice 14 of arbiter.
23:20	Arb13	RW 0xd	Slice 13 of arbiter.
19:16	Arb12	RW 0xc	Slice 12 of arbiter.
15:12	Arb11	RW 0xb	Slice 11 of arbiter.
11:8	Arb10	RW 0xa	Slice 10 of arbiter.
7:4	Arb9	RW 0x9	Slice 9 of arbiter.
3:0	Arb8	RW 0x8	Slice 8 of arbiter.

Table 524: PCI Express Mbus Arbiter Timeout Register (p=1–3)
Offset: pcie_port_num1: 0x000418E8, pcie_port_num2: 0x000458E8, pcie_port_num3: 0x000498E8
Offset Formula: 0x000418E8+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	TimeoutEn	RW 0x1	Mbus Arbiter Timeout Timer Enable 0 = Enable 1 = Disable
15:8	Reserved	RSVD 0x0	Reserved
7:0	Timeout	RW 0xFF	Mbus Arbiter Timeout Preset Value **NOTE:** Must keep the value of 0xFF as long as the Timeout counter is not enabled.

A.8.5 PCI Express Control and Status (Ports 1-3)

Table 525: PCI Express Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041A00, pcie_port_num2: 0x00045A00, pcie_port_num3: 0x00049A00
Offset Formula: 0x00041A00+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31	CRS Enable	RW 0x0	When this bit is set, all configuration access returns with status=<CRS>. 0 = Disable 1 = Enable

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 525: PCI Express Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A00, pcie_port_num2: 0x00045A00, pcie_port_num3: 0x00049A00
Offset Formula: 0x00041A00+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
30	Conf Training Disable	RW 0x1	MAC Training Disable
29	Reserved	RSVD 0x0	Reserved
28	ConfMove2DetectFromModCompl	RW 0x0	If asserted, causes link restart while in Polling.MOD.COMPL state.
27	ConfMstrDis Scrmb	RW 0x0	Master Disable Scrambling When set, a scrambling disable command is transmitted downstream, causing the scrambling to be disabled on both sides of the link. **NOTE:** This should be programmed before the start of link initialization.
26	ConfMstrLb	RW 0x0	Master Loopback When set, a loopback command is transmitted downstream, causing the downstream device to transmit back the traffic that it receives. This bit can be set only if bits[24] of this register and <LnkDis> in the PCI Express Link Control Status Register are cleared. **NOTE:** Use this bit only when working in Root Complex mode.
25	Compliance Receive	RW 0x0	Compliance Receive When set, a compliance command (Bit 4 of Symbol 5 in TS1 OS) is asserted in TS1 OS, causing the device connected to move to Polling.Compliance state.
24	ConfMstrHot Reset	RW 0x0	Master Hot-Reset When set, a hot-reset command is transmitted downstream, causing the downstream PCI Express hierarchy to enter a hot-reset cycle. This bit can be set only if LnkDis in the PCI Express Link Control Status Register and bit[26] of this register are cleared. Activation procedure: 1. Set this bit to trigger a hot-reset cycle. 2. Poll DLDown de-assertion (PCI Express Status Register, bit 0) to ensure hot-reset cycle is done. 3. Clear this bit to exit to detect and re-establish the PCI-Express link. This bit should be used only when working in Root Complex mode.
23:16	Reserved	RW 0x14	Reserved Must write 0x14. **NOTE:**
15:13	Reserved	RSVD 0x0	Reserved
12	Reserved	RW 0x0	Reserved
11	Reserved	RW 0x0	Reserved

Table 525: PCI Express Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A00, pcie_port_num2: 0x00045A00, pcie_port_num3: 0x00049A00
Offset Formula: 0x00041A00+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
10	conf_auto_speed_change	RW 0x0	Auto Speed change. when set , link will issue link speed change to the max link speed possibel. 0x0 do not initiate a change. 0x1 do initiate a change.
9:8	Reserved_9_8	RW SAR	Reserved Please do not change default value at any time. The default value = 0x2
7:3	Reserved	RSVD 0x5	Reserved
2	CfgMapTo MemEn	RW 0x1	Configuration Header Mapping to Memory Space Enable When enabled, the configuration header registers can be accessed directly through the memory space. Access is enabled both from the internal bus and from the PCI Express port. 0 = Disable: Mapping disabled. 1 = Enable: Mapping enabled.
1	ConfRoot Complex	RW SAR	PCI Express Device Type- this field is not in use in PCIe ports 0 and 1 and should be ignored 0 = Endpoint: Endpoint device (Downstream device / Upstream link). 1 = Root: Root Complex device (Upstream device / Downstream link).
0	Reserved	RW 0x0	Reserved

Table 526: PCI Express Status Register (p=1–3)
Offset: pcie_port_num1: 0x00041A04, pcie_port_num2: 0x00045A04, pcie_port_num3: 0x00049A04
Offset Formula: 0x00041A04+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27	PCIeSlvDis Scrmb	RO 0x0	Slave Disable Scrambling Indication This field sets when the opposite device on the PCI Express port acts as a disable scrambling master, and when a scrambling disabled indication is received.
26	PCIeSlvLb	RO 0x0	Slave Loopback Indication This field sets when the opposite device on the PCI Express port acts as a loopback master, and when a loopback indication is received.
25	PCIeSlvDisLink	RO 0x0	Slave Disable Link Indication This field sets when the opposite device on the PCI Express port acts as a disable link master, and when a link disabled indication is received.
24	PCIeSlvHot Reset	RO 0x0	Slave Hot Reset Indication This field sets when the opposite device on the PCI Express port acts as a hot-reset master, and a when hot-reset indication is received.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 526: PCI Express Status Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A04, pcie_port_num2: 0x00045A04, pcie_port_num3: 0x00049A04
Offset Formula: 0x00041A04+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
23:21	Reserved	RSVD 0x0	Reserved
20:16	PCIeDevNum	RW 0x0	Device Number Indication This field is used in the RequesterID field of the transmitted TLPs. In Endpoint mode, the field updates whenever a CfgWr access is received.
15:8	PCIeBusNum	RW 0x0	Bus Number Indication This field is used in the RequesterID field of the transmitted TLPs. In Endpoint mode, the field updates whenever a CfgWr access is received.
7:5	Reserved	RSVD 0x0	Reserved
4:1	Reserved	RSVD 0x0	Reserved
0	DLDown	RO 0x0	DL_Down Indication 0 = Active: DL is up. 1 = Inactive: DL is down.

Table 527: PCI Express Spare Register (p=1–3)
Offset: pcie_port_num1: 0x00041A08, pcie_port_num2: 0x00045A08, pcie_port_num3: 0x00049A08
Offset Formula: 0x00041A08+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0xFFFF0007	Reserved

Table 528: PCI Express Root Complex Set Slot Power Limit (SSPL) Register (p=1–3)
Offset: pcie_port_num1: 0x00041A0C, pcie_port_num2: 0x00045A0C, pcie_port_num3: 0x00049A0C
Offset Formula: 0x00041A0C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:17	Reserved	RO 0x0	Reserved
16	SsplMsgEnable	RW 0x0	SSPL Message Enable 0 = Disable: No SSPL message is send. 1 = Enable: SSPL message is send when the PCI Express is set to Root Complex and: 1) DI-Down -> DI-Up or 2) There is a Write to the <SlotPowerLimitValue> or <SlotPowerLimitScale> fields.
15:10	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 528: PCI Express Root Complex Set Slot Power Limit (SSPL) Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A0C, pcie_port_num2: 0x00045A0C, pcie_port_num3: 0x00049A0C
Offset Formula: 0x00041A0C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
9:8	SlotPowerLimitScale	RW 0x0	Slot Power Limit Scale Specifies the scale used for the Slot Power Limit Value (in the PCI Express Interface section see the sub-section "Messages"). Range of Values: 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x Writes to this register when the PCI Express interface is set to Root Complex also cause the port to send the Set Slot Power Limit message.
7:0	SlotPowerLimitValue	RW 0x0	Slot Power Limit Value In combination with the <SlotPowerLimitScale> value, specifies the upper limit on power supplied by the slot. The power limit (in watts) calculated by multiplying the value in this field by the value in the <SlotPowerLimitScale> field. Writes to this register when the PCI Express interface is set to Root Complex also cause the port to send the Set Slot Power Limit message.

Table 529: PCI Express Completion Timeout Register (p=1–3)
Offset: pcie_port_num1: 0x00041A10, pcie_port_num2: 0x00045A10, pcie_port_num3: 0x00049A10
Offset Formula: 0x00041A10+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	CPL To Range Offset	RW 0x0	Completion Range Offset This field is used to select a value from the range indicated in the <CplTOValue> field in the PCI Express Control Status 2 register. For example, the <CplTOValue> field provides a default range: 50 us to 50 ms and this register indicates whether the value is close to the minimum or maximum value in this range. 0 = Min Value: Minimum value in the selected range. 1 = Max Value: Maximum value in the selected range.
15:0	Reserved	RW 0x2710	Reserved

Table 530: PCI Express Root Complex Power Management Event (PME) Register (p=1–3)
Offset: pcie_port_num1: 0x00041A14, pcie_port_num2: 0x00045A14, pcie_port_num3: 0x00049A14
Offset Formula: 0x00041A14+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:18	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 530: PCI Express Root Complex Power Management Event (PME) Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A14, pcie_port_num2: 0x00045A14, pcie_port_num3: 0x00049A14
Offset Formula: 0x00041A14+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
17	PME Pending	RO 0x0	PME Pending This read-only bit indicates that another PME is pending when the PME Status bit is set. When the PME Status bit is cleared by software, the PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending.
16	PME Status	RO 0x0	PME Status This bit indicates that PME was asserted by the requestor ID indicated in the PME Requestor ID field. This bit can be cleared by a write of 0x1 to the <RcvPmPme> field in the PCI Express Interrupt Cause register (0x1900[28]).
15:0	PME Requester ID	RO 0x0	PME Requestor ID This field indicates the PCI requestor ID of the last locked PME requestor. When PME_Status=0x0, the value in the PME_Requester_ID is regard as reserved.

Table 531: PCI Express Power Management Register (p=1–3)
Offset: pcie_port_num1: 0x00041A18, pcie_port_num2: 0x00045A18, pcie_port_num3: 0x00049A18
Offset Formula: 0x00041A18+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:17	Reserved	RW 0x0	Reserved
16	Ref_clk_off_en	RW 0x0	The system enables the hardware to kill the REFCLK and Core Clock. This input is also a condition for gating the PCI Express unit.
15:6	Reserved	RSVD 0x0	Reserved
5	Send Turn Off Msg	RW 0x0	Host Request PCI Express unit to send Turn Off message.
4	Send Turn Off Ack Msg	RW 0x0	Endpoint Ready for Turn Off and ready to send Turn Off ack message and start sending DLLP Enter L2/3.
3:2	Reserved	RSVD 0x0	Reserved
1	L1 ASPM Ack	RW 0x1	Only for Root Complex. The system allows acknowledgment of an L1 ASPM request by an Endpoint. 0 = SendAspmNackTlp: Stop L1-ASPM flow. 1 = SendAspmAckDllp: Continue L1-ASPM flow.
0	L1 ASPM En	RW 0x0	Only for Endpoint. Enables the hardware to initialize the L1 ASPM process. 0 = IDLE: No l1-Aspm flow (as EndPoint) 1 = StartL1AspmFlow

Copyright © 2015 Marvell

Table 532: PCI Express Flow Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041A20, pcie_port_num2: 0x00045A20, pcie_port_num3: 0x00049A20
Offset Formula: 0x00041A20+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31	Disable_UpdateFC_TO	RW 0x0	Disable Update Flow Control due to 30 usec timeout. 0 = EnableUpdateFC: There will be at least one Update Flow Control each 30 usec. 1 = DisableUpdateFC: There will be no Update Flow Control due to 30 usec timer expiration.
30:24	Reserved	RW 0x0	Reserved
23:16	ConfChInitFc	RW 0x0	Completion Headers Flow Control Credit Initial Value Infinite.
15:8	ConfNphInitFc	RW SAR	Non-Posted Headers Flow Control Credit Initial Value The default value = 0x8
7:0	ConfPhInitFc	RW SAR	Posted Headers Flow Control Credit Initial Value The default value = 0x4

Table 533: PCI Express Replay Timeout (TO) Timer Register (p=1–3)
Offset: pcie_port_num1: 0x00041A24, pcie_port_num2: 0x00045A24, pcie_port_num3: 0x00049A24
Offset Formula: 0x00041A24+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:10	Reserved	RSVD 0x0	Reserved
9	Disable Replay TO	RW 0x0	Disable Replay TO 1 = No Replay TO: There will be no Replay TO.
8	Use Change Replay TO	RW 0x0	Use Change Replay TO 0 = Fixed: The design value is taken from the fixed initial value. 1 = Bit[7:0] Value: The design value is taken from the <Change Replay TO> bits[7:0].
7:0	Change Replay TO	RW 0x0	Change Replay TO value used when <Use Change Replay TO> is asserted.

Table 534: PCI Express Ack Latency Timer Register (p=1–3)
Offset: pcie_port_num1: 0x00041A28, pcie_port_num2: 0x00045A28, pcie_port_num3: 0x00049A28
Offset Formula: 0x00041A28+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 534: PCI Express Ack Latency Timer Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A28, pcie_port_num2: 0x00045A28, pcie_port_num3: 0x00049A28
Offset Formula: 0x00041A28+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
8	Use Change Ack TO	RW 0x0	Use Change Ack TO 0 => The design value is taken from fixed init-val. 1 => The design value is taken from change_ack_to[7:0]. 0 = Fixed: The design value is taken from fixed initial value. 1 = Bit[7:0] Value: The design value is taken from the <Use Change Ack TO> field bits[7:0].
7:0	Change Ack TO	RW 0x0	The value used for ACT_TO when the <Use Change Ack TO> field is asserted.

Table 535: PCI Express Secondary Bus Register (p=1–3)
Offset: pcie_port_num1: 0x00041A2C, pcie_port_num2: 0x00045A2C, pcie_port_num3: 0x00049A2C
Offset Formula: 0x00041A2C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:9	Reserved	RO 0x0	
8	Secondary Bus Number Enable	RW 0x0	When 1, the CFG TLP TYPE is defined using Secondary Bus number. Otherwise the Primary bus number is used.
7:0	Secondary Bus Number	RW 0x0	The Secondary Bus number field is used to decide on type on CFG request. If the Destination Bus number equals the Secondary Bus number then type 0; otherwise type 1. This field is used if the <Secondary Bus Number Enable> field is 1'b1; otherwise the Primary Bus is used for the type calculation.

A.8.6 PCI Express BAR Control (Ports 1-3)

Table 536: PCI Express BAR1 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041804, pcie_port_num2: 0x00045804, pcie_port_num3: 0x00049804
Offset Formula: 0x00041804+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Bar1Size	RW SAR	BAR1 Size BAR sizes range from 64 KByte to 4 GByte in powers of 2. Default value: 03FFF = 1 GByte
15:1	Reserved	RSVD 0x0	Reserved
0	Bar1En	RW 0x1	BAR1 Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 537: PCI Express BAR2 Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041808, pcie_port_num2: 0x00045808, pcie_port_num3: 0x00049808
Offset Formula: 0x00041808+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:16	Bar2Size	RW SAR	BAR2 Size BAR sizes range from 64 KByte to 4 GByte in powers of 2. Default value: 0x0FFF = 256 MByte
15:1	Reserved	RSVD 0x0	Reserved
0	Bar2En	RW 0x1	BAR2 Enable 0 = Disable 1 = Enable

Table 538: PCI Express Expansion ROM BAR Control Register (p=1–3)
Offset: pcie_port_num1: 0x0004180C, pcie_port_num2: 0x0004580C, pcie_port_num3: 0x0004980C
Offset Formula: 0x0004180C+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved
21:19	ExpROMSz	RW 0x0	Expansion ROM Address Bank Size 0 = 512 KByte 1 = 1024 KByte 2 = 2048 KByte 3 = 4096 KByte
18:1	Reserved	RSVD 0x0	Reserved
0	ExpROMEn	RW 0x0	Expansion ROM BAR Enable 0 = Disable 1 = Enable

A.8.7 PCI Express Interrupt (Ports 1-3)

Table 539: PCI Express Interrupt Cause Register (p=1–3)
Offset: pcie_port_num1: 0x00041900, pcie_port_num2: 0x00045900, pcie_port_num3: 0x00049900
Offset Formula: 0x00041900+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
All bits, except bits[27:24], are RW0C only. A cause bit sets upon an event occurrence. A write of 0 clears the bit. A write of 1 has no effect. Bits[24:27] are set and cleared upon reception of interrupt emulation messages.			
31	RcvMsi	RW0C 0x0	RC received MemWr that hit the MSI attributes.
30	Reserved	RW0C 0x0	Reserved

Document Classification: Proprietary Information

Table 539: PCI Express Interrupt Cause Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041900, pcie_port_num2: 0x00045900, pcie_port_num3: 0x00049900
Offset Formula: 0x00041900+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
29	RcvTurnOff	RW0C 0x0	If PCIe is Endpoint, RcvTurnOffMsg: Receive Turn Off MSG from the RC. If PCIe is RootComplex, Ready4TurnOff: RC notifies the Host that the Turn Off Process is completed (whether normally or by a timer).
28	RcvPmPme	RW0C 0x0	Received PM_PME message.
27	RcvIntD	RO 0x0	IntDn status Reflects IntD Interrupt message emulation status. Set when IntD_Assert message received. Cleared when IntD_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
26	RcvIntC	RO 0x0	IntC status Reflects IntC Interrupt message emulation status. Set when IntC_Assert message received. Cleared when IntC_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
25	RcvIntB	RO 0x0	IntB status Reflects IntB Interrupt message emulation status. Set when IntB_Assert message received. Cleared when IntB_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
24	RcvIntA	RO 0x0	IntA status Reflects IntA Interrupt message emulation status. Set when IntA_Assert message received. Cleared when IntA_Deassert message received, or upon link failure scenario (Dl_down). **NOTE:** This bit is not RW0C as other bits in this register, since it is cleared by the interrupting device downstream. Relevant for Root Complex only.
23	PCIeLinkFail	RW0C 0x0	Link Failure indication PCI Express link dropped from active state (L0, L0s or L1) to Detect state due to link errors. **NOTE:** When dropping to Detect via Hot Reset, Disable Link or Loopback states, the interrupt is not asserted. Sticky bit--not initialized by reset.
22	PCIeSlvLb	RW0C 0x0	Slave Loopback Indication The bit sets when the opposite device on the PCI Express port is acting as a loopback master, and loopback mode was entered. **NOTE:** Sticky bit--not initialized by reset.

Table 539: PCI Express Interrupt Cause Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041900, pcie_port_num2: 0x00045900, pcie_port_num3: 0x00049900
Offset Formula: 0x00041900+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
21	PCIeSlvDisLink	RW0C 0x0	Slave Disable Link Indication The bit sets when the opposite device on the PCI Express port is acting as a disable link master, and link was disabled. **NOTE:** Sticky bit--not initialized by reset.
20	PCIeSlvHot Reset	RW0C 0x0	Received Hot Reset Indication The bit sets when a hot-reset indication is received from the opposite device on the PCI Express port. **NOTE:** Sticky bit--not initialized by reset.
19	RcvCRS	RW0C 0x0	Received CRS completion status. A downstream PCI Express device can respond to a configuration request with CRS (Configuration Request Retry Status) if it is not ready yet to serve the request. RcvCRS interrupt is set when such a completion status is received.
18	RcvErrCor	RW0C 0x0	Received ERR_COR message. Set when a downstream device detected the error and sent an error message upstream. Relevant for Root Complex only.
17	RcvErrNonFatal	RW0C 0x0	Received ERR_NONFATAL message. Set when a downstream device detected the error and sent an error message upstream. Relevant for Root Complex only.
16	RcvErrFatal	RW0C 0x0	Received ERR_FATAL message. Set when a downstream device detected the error and sent an error message upstream. Relevant for Root Complex only.
15	RcvUrCaErr	RW0C 0x0	Received either UR or CA completion status.
14	FlowCtrlProtocol	RW0C 0x0	Flow Control Protocol Error Set when there is no FC DLLP for more than 200 usec.
13	Reserved	RW0C 0x0	Reserved
12	BIST	RW0C 0x0	PCI Express BIST Enable BIST is not supported.
11	DstateChange	RW0C 0x0	Dstate Change Indication Any change in the Dstate asserts this bit. **NOTE:** This bit is relevant only for Endpoint.
10	FErrDet	RW0C 0x0	Fatal Error Detected Indicates status of Fatal errors detected by the device.
9	NFErrDet	RW0C 0x0	Non-Fatal Error Detected Indicates status of Non-Fatal errors detected by the device.
8	CorErrDet	RW0C 0x0	Correctable Error Detected Indicates status of correctable errors detected by the device.

Copyright © 2015 Marvell

Table 539: PCI Express Interrupt Cause Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041900, pcie_port_num2: 0x00045900, pcie_port_num3: 0x00049900
Offset Formula: 0x00041900+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
7	TxRamParErr	RW0C 0x0	Tx RAM Parity Error
6	RxRamParErr	RW0C 0x0	Rx RAM Parity Error
5	Reserved	RW0C 0x0	Reserved
4	HitDfltWinErr	RW0C 0x0	Hit Default Window Error
3	ErrWrToReg	RW0C 0x0	Erroneous write attempt to internal register. Set when an erroneous write request to PCI Express internal register is received, either from the PCI Express (EP set) or from the internal bus (bit[64] set).
2	Reserved	RW0C 0x0	Reserved
1	MDis	RW0C 0x0	Attempt to generate a PCI transaction while master is disabled.
0	TxReqInDldownErr	RW0C 0x0	Receive TxReq while PCI Express is in Dl-Down.

Table 540: PCI Express Interrupt Mask Register (p=1–3)
Offset: pcie_port_num1: 0x00041910, pcie_port_num2: 0x00045910, pcie_port_num3: 0x00049910
Offset Formula: 0x00041910+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:24	Mask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the interrupt.
23:20	Mask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the interrupt.
19:0	Mask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the interrupt.

Copyright © 2015 Marvell

A.8.8 PCI Express Debug (Ports 1-3)

Table 541: PCI Express Debug Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041A60, pcie_port_num2: 0x00045A60, pcie_port_num3: 0x00049A60
Offset Formula: 0x00041A60+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x3D	Reserved
21	DisHotResetRegRst	RW 0x1	Disable PCIe Register File reset upon hot reset. **NOTE:** Sticky bit -- not initialized by reset. If set to 0x0 (reset is enabled), Regfile initialization is necessary again, if it overwrites the PCIe configurations. 0 = Disable: Mask Disable 1 = Enable: Mask Enable
20	Reserved	RSVD 0x0	Reserved
19	DisLinkRestartRegRst	RW 0x1	Disable PCIe Register File reset upon Link restart (as a result of link disable prosedure or link instability) Note: Sticky bit - not initialized by reset 0 = Disable: Mask Disable 1 = Enable: Mask Enable
18	Reserved	RSVD 0x0	Reserved
17	ConfMskHotReset	RW 0x1	Mask Hot Reset from PCIe Reset Output **NOTE:** Sticky bit -- not initialized by reset. 0 = Disable: Mask Disable 1 = Enable: Mask Enable
16	ConfMskLnkRestart	RW 0x1	Mask Reset propagation to PCIE Reset Output in case of link restart event (as a result of link disable prosedure or link instability) Note: Sticky bit - not initialized by reset 0 = Disable: Mask Disable 1 = Enable: Mask Enable
15:0	Reserved	RSVD 0xF0C0	Reserved

Table 542: PCI Express Debug MAC Status 1 Register (p=1–3)
Offset: pcie_port_num1: 0x00041A84, pcie_port_num2: 0x00045A84, pcie_port_num3: 0x00049A84
Offset Formula: 0x00041A84+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:24	TrnCntlBits	RO 0x0	Training control status bits sent on TX: Bit 27 - Scramble disable Bit 26 - Master loopback Bit 25 - Link disable Bit 24 - Hot reset

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 542: PCI Express Debug MAC Status 1 Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041A84, pcie_port_num2: 0x00045A84, pcie_port_num3: 0x00049A84
Offset Formula: 0x00041A84+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
23:0	Reserved	RSVD 0x0	Reserved

Table 543: PCI Express TL Control Register (p=1–3)
Offset: pcie_port_num1: 0x00041AB0, pcie_port_num2: 0x00045AB0, pcie_port_num3: 0x00049AB0
Offset Formula: 0x00041AB0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
31:14	Reserved	RSVD 0x0	Reserved
13:10	PDedicateBufferSize	RW 0x3	Indicates the number of entries used for posted tlps allocation. The maximum number for X1 mode is 5 , for X4 mode is 11. The number of CPL entries is calculated as 6 - <PDedicateBufferSize> for X1 mode and 12 - <PDedicateBufferSize> for X4 mode.
9	PCplDedicateAlloc	RW 0x0	If asserted, indicates that the dynamic buffer allocation is disabled and the dedicated buffer sizes are used for completion (CPL) and posted (P) buffers. 0 = Dynamic allocation: Dynamic allocation is enabled. 1 = Dedicated allocation: CPL and P tlps each have their own resources.
8	UpdateDedicatedBufferAlloc	RW 0x0	To latch the allocation of Posted and Completion buffers from the <PDedicateBufferSize> field of current register, the user must write 1’b1 to this field. Otherwise, the design will assume that the size of Completion and Posted buffers are equal. To update the size of buffers again, without initiating a fundamental reset, write 1’b0 to this field and then 1’b1 again.
7	Reserved	RSVD 0x0	Reserved
6	Reserved	RW 0x0	Reserved
5	Reserved	RW 0x0	Reserved
4	Reserved	RW 0x0	Reserved
3	TxNpPushDis	RW 0x0	Tx Transaction Ordering Control (Non Posted vs. Posted) 0 = Disable 1 = Enable: Tx non posted may bypass posted.
2	TxCmplPushDis	RW 0x0	Tx Transaction Ordering Control (Completion vs. Posted) 0 = Disable 1 = Enable: Tx completion may bypass posted.
1	RxNpPushDis	RW 0x0	Rx Transaction Ordering Control (Non Posted vs. Posted) 0 = Disable 1 = Enable: Rx non posted may bypass posted.

Copyright © 2015 Marvell

Table 543: PCI Express TL Control Register (p=1–3) (Continued)
Offset: pcie_port_num1: 0x00041AB0, pcie_port_num2: 0x00045AB0, pcie_port_num3: 0x00049AB0
Offset Formula: 0x00041AB0+0x4000*(p-1): where p (1-3) represents pcie_port_num

Bit	Field	Type/InitVal	Description
0	RxCmplPushDis	RW 0x0	Rx Transaction Ordering Control (Completion vs. Posted) 0 = Disable 1 = Enable: Rx completion may bypass posted.

Copyright © 2015 Marvell

A.9 USB 2.0 Registers

The following table provides a summarized list of all registers that belong to the USB 2.0, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 544: Summary Map Table for the USB 2.0 Registers

Register Name	Offset	Table and Page
USB 2.0 Bridge Interrupt and Error Registers		
USB 2.0 Bridge Interrupt Cause Register	0x00058310	Table 545, p. 691
USB 2.0 Bridge Interrupt Mask Register	0x00058314	Table 546, p. 692
USB 2.0 Core Interrupt Mask Register	0x00058318	Table 547, p. 692
USB 2.0 Bridge Error Address Register	0x0005831C	Table 548, p. 692
USB 2.0 Bridge Control and Status Registers		
USB 2.0 Bridge Control Register	0x00058300	Table 549, p. 693
USB 2.0 Bridge Address Decoding Registers		
USB 2.0 Window0 Control Register	0x00058320	Table 550, p. 693
USB 2.0 Window0 Base Register	0x00058324	Table 551, p. 694
USB 2.0 Window1 Control Register	0x00058330	Table 552, p. 694
USB 2.0 Window1 Base Register	0x00058334	Table 553, p. 695
USB 2.0 Window2 Control Register	0x00058340	Table 554, p. 695
USB 2.0 Window2 Base Register	0x00058344	Table 555, p. 695
USB 2.0 Window3 Control Register	0x00058350	Table 556, p. 696
USB 2.0 Window3 Base Register	0x00058354	Table 557, p. 696
USB 2.0 PHY Registers		
USB 2.0 Power Control Register	0x00058400	Table 558, p. 696
USB 2.0 PHY Suspend/Resume Control Register	0x00058404	Table 559, p. 697

A.9.1 USB 2.0 Bridge Interrupt and Error Registers

Table 545: USB 2.0 Bridge Interrupt Cause Register
Offset: 0x00058310

Bit	Field	Type/InitVal	Description
NOTE: All cause bits are clear only. They are set to 1 upon an interrupt event and cleared when the software writes a value of 0. Writing 1 has no affect.			
31:7	Reserved	RO 0x0	Reserved
6:4	Reserved	RW0C 0x0	Reserved

Table 545: USB 2.0 Bridge Interrupt Cause Register (Continued)
Offset: 0x00058310

Bit	Field	Type/InitVal	Description
3	DUF	RW0C 0x0	USB Device underflow
2	DOF	RW0C 0x0	USB Host/Device overflow
1	HOF	RW0C 0x0	USB Host Underflow
0	AddrDecErr	RW0C 0x0	Address Decoding Error Asserted upon address decoding error

Table 546: USB 2.0 Bridge Interrupt Mask Register
Offset: 0x00058314

Bit	Field	Type/InitVal	Description
31:7	Reserved	RSVD 0x0	Reserved
6:4	Reserved	RW 0x0	Reserved
3:0	Mask	RW 0x0	If set to 1, the related interrupt is enabled.

Table 547: USB 2.0 Core Interrupt Mask Register
Offset: 0x00058318

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	UsbIntrMask	RW 0x1	When set to 1, USB port interrupts are routed the CPU core. If set to 0, USB port interrupts are not routed to the CPU core. 0 = Mask 1 = Not_Mask

Table 548: USB 2.0 Bridge Error Address Register
Offset: 0x0005831C

Bit	Field	Type/InitVal	Description
31:0	ErrAddr	RO 0x0	Error Address Latched upon any of the address decoding errors (address miss, multiple hit). Once the address is latched, no new address is latched until SW reads it (Read access to USB 2.0 Bridge Error Address Register).

Copyright © 2015 Marvell

A.9.2 USB 2.0 Bridge Control and Status Registers

Table 549: USB 2.0 Bridge Control Register
Offset: 0x00058300

Bit	Field	Type/InitVal	Description
31:10	Reserved	RSVD 0x0	Reserved
9:6	Reserved	RW 0x0	Reserved
5	Reserved	RW 0x0	Reserved 0 = reserved0 1 = reserved1
4	BS	RW 0x1	USB Core Byte Swap 0 = Byte swap: Byte swap over 64-bit qword, upon USB core read/write access from memory. 1 = No byte swap
3	Reserved	RW 0x0	Reserved
2	Reserved	RW 0x0	Reserved 0 = reserved0 1 = reserved1
1	Reserved	RW 0x0	0 = reserved0 1 = reserved1
0	Reserved	RW 0x0	Reserved 0 = reserved0 1 = reserved1

A.9.3 USB 2.0 Bridge Address Decoding Registers

Table 550: USB 2.0 Window0 Control Register
Offset: 0x00058320

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0's. The number of 1s specifies the size of the window (for example, a value of 0x00FF specifies 256 x 64k = 16 MB).
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this window. **NOTE:** See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. See the Units IDs and Attributes table. **NOTE:** See the "PCI Express Address Decoding" section. Must be configured to the SDRAM Controller.
3:2	Reserved	RSVD 0x0	Reserved

Table 550: USB 2.0 Window0 Control Register (Continued)
Offset: 0x00058320

Bit	Field	Type/InitVal	Description
1	WrBL	RW 0x0	USB to Mbus Burst Write Limit 0 = No limit 1 = Limit: Limit writes not to cross 32B boundary
0	win_en	RW 0x0	Window0 Enable 0 = Disable 1 = Enable

Table 551: USB 2.0 Window0 Base Register
Offset: 0x00058324

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0	Base Address Used with the size field to set the address window size and location. Corresponds to transaction address[31:16]
15:0	Reserved	RSVD 0x0	Reserved

Table 552: USB 2.0 Window1 Control Register
Offset: 0x00058330

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window (for example, a value of 0x00FF specifies 256x64k = 16 MB).
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this window. **NOTE:** See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. See the Units IDs and Attributes table. **NOTE:** See the "PCI Express Address Decoding" section. Must be configured to the SDRAM Controller.
3:2	Reserved	RSVD 0x0	Reserved
1	WrBL	RW 0x0	USB to Mbus Burst Write Limit 0 = No limit: No limit 1 = Limit: Limit writes not to cross 32B boundary
0	win_en	RW 0x0	Window1 Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 553: USB 2.0 Window1 Base Register
Offset: 0x00058334

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0	Base Address Used with the size field to set the address window size and location. Corresponds to transaction address[31:16]
15:0	Reserved	RSVD 0x0	Reserved

Table 554: USB 2.0 Window2 Control Register
Offset: 0x00058340

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window (for example, a value of 0x00FF specifies 256x64k = 16 MB).
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this windo. **NOTE:** See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. See the Units IDs and Attributes table. **NOTE:** See the "PCI Express Address Decoding" section. Must be configured to the SDRAM Controller.
3:2	Reserved	RSVD 0x0	Reserved
1	WrBL	RW 0x0	USB to Mbus Burst Write Limit 0 = No limit 1 = Limit: Limit writes not to cross 32B boundary
0	win_en	RW 0x0	Window2 Enable 0 = Disable 1 = Enable

Table 555: USB 2.0 Window2 Base Register
Offset: 0x00058344

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0	Base Address Used with the size field to set the address window size and location. Corresponds to transaction address[31:16]
15:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 556: USB 2.0 Window3 Control Register
Offset: 0x00058350

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1's specifies the size of the window (for example, a value of 0x00ff specifies 256x64k = 16 MB).
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this window. **NOTE:** See the "PCI Express Address Decoding" section.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. See the Units IDs and Attributes table. **NOTE:** See the "PCI Express Address Decoding" section. Must be configured to the SDRAM Controller.
3:1	Reserved	RSVD 0x0	Reserved
0	win_en	RW 0x0	Window3 Enable 0 = Disable 1 = Enable

Table 557: USB 2.0 Window3 Base Register
Offset: 0x00058354

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0	Base Address Used with the size field to set the address window size and location. Corresponds to transaction address[31:16]
15:0	Reserved	RSVD 0x0	Reserved

A.9.4 USB 2.0 PHY Registers

Table 558: USB 2.0 Power Control Register
Offset: 0x00058400

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x1FE0002C	Reserved
2	SUSPENDM	RW 0x1	Input SUSPENDM Note:This bit is valid only when bit 1 in register suspend/resume control (0x404) is set to 0x1. 0 = Force PHY to suspend 1 = Normal operation
1:0	Reserved	RSVD 0x3	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 559: USB 2.0 PHY Suspend/Resume Control Register
Offset: 0x00058404

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x0	Reserved
1	Enable force suspend	RW 0x0	Force Suspend Enable 0 = Disable force suspend 1 = Enable force suspend: (See PHY Power Control register bit[2])
0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.10 USB2 PHY (UTMI PHY) Registers

The following table provides a summarized list of all registers that belong to the USB2 PHY (UTMI PHY) Registers, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 560: Summary Map Table for the USB2 PHY (UTMI PHY) Registers Registers

Register Name	Offset	Table and Page
USB2 PHY		
PLL Control 0 Register (n=0–2)	Port0: 0x000C0000, Port1: 0x000C2000, Port2: 0x000C4000	Table 561, p. 699
PLL Control 1 Register (n=0–2)	Port0: 0x000C0004, Port1: 0x000C2004, Port2: 0x000C4004	Table 562, p. 700
Calibration Control Register (n=0–2)	Port0: 0x000C0008, Port1: 0x000C2008, Port2: 0x000C4008	Table 563, p. 701
TX Channel Control Register 0 (n=0–2)	Port0: 0x000C000C, Port1: 0x000C200C, Port2: 0x000C400C	Table 564, p. 703
TX Channel Control Register 1 (n=0–2)	Port0: 0x000C0010, Port1: 0x000C2010, Port2: 0x000C4010	Table 565, p. 705
Rx Channel Control 0 Register (n=0–2)	Port0: 0x000C0014, Port1: 0x000C2014, Port2: 0x000C4014	Table 566, p. 705
Rx Channel Control 1 Register (n=0–2)	Port0: 0x000C0018, Port1: 0x000C2018, Port2: 0x000C4018	Table 567, p. 707
Digital Control Register 0 (n=0–2)	Port0: 0x000C001C, Port1: 0x000C201C, Port2: 0x000C401C	Table 568, p. 708
Digital Control Register 1 (n=0–2)	Port0: 0x000C0020, Port1: 0x000C2020, Port2: 0x000C4020	Table 569, p. 710
Test Control and Status Register 0 (n=0–2)	Port0: 0x000C0024, Port1: 0x000C2024, Port2: 0x000C4024	Table 570, p. 712
Test Control and Status Register 1 (n=0–2)	Port0: 0x000C0028, Port1: 0x000C2028, Port2: 0x000C4028	Table 571, p. 714
Monitor Register (n=0–2)	Port0: 0x000C002C, Port1: 0x000C202C, Port2: 0x000C402C	Table 572, p. 714
PHY RESERVE Register (n=0–2)	Port0: 0x000C0030, Port1: 0x000C2030, Port2: 0x000C4030	Table 573, p. 714

Document Classification: Proprietary Information

Table 560: Summary Map Table for the USB2 PHY (UTMI PHY) Registers Registers (Continued)

Register Name	Offset	Table and Page
PHY Charger Detector Control Register (n=0–2)	Port0: 0x000C0038, Port1: 0x000C2038, Port2: 0x000C4038	Table 574, p. 715
Reserved. Register (n=0–2)	Port0: 0x000C003C, Port1: 0x000C203C, Port2: 0x000C403C	Table 575, p. 716

A.10.1 USB2 PHY

Table 561: PLL Control 0 Register (n=0–2)
Offset: Port0: 0x000C0000, Port1: 0x000C2000, Port2: 0x000C4000
Offset Formula: 0x000C0000+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31	PLL_READY	RO 0x0	PLL Ready 0 = PLL not locked 1 = PLL locked
30	R_ROTATE	RW 0x1	Rotate Select 0 = control is rotated by 90 degrees 1 = reference resistor is parallel with the pad
29:28	SEL_LPFR	RW 0x0	Select LPFR Set to 00 if frefclk/REFDIV[6:0] = 4MHz to 5MHz range Set to 01 if frefclk/REFDIV[6:0] is in 2MHz to 2.5MHz range Set to 10 if frefclk/REFDIV[6:0] is in 1MHz to 1.25MHz range 3 = Reserved
27:25	Reserved_bit27_25	RW 0x0	
24:16	FBDIV	RW 0x78	Feedback Clock Divider Select Default is set for 40 MHz REFCLK. If a REFCLK frequency other than 40 MHz is used, this value must be changed before powering up the PLL. 0 = 0x0F = illegal 16 = 0x1FF = divided by 16 to 511 directly
15:14	VDDL	RW 0x1	PLL 1.1V Regulator Output Voltage Control 0 = VDD 1 = 0.9V 2 = 1.0V 3 = 1.1V

Table 561: PLL Control 0 Register (n=0–2) (Continued)
Offset: Port0: 0x000C0000, Port1: 0x000C2000, Port2: 0x000C4000
Offset Formula: 0x000C0000+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
13:12	VDDM	RW 0x1	1.8V Regulator Voltage Control 0 = 1.7V 1 = 1.8V 2 = 1.9V 3 = 2.0V
11:8	ICP	RW 0x2	PLL Charge-Pump Current Select ICP[3:0]: ICP 0 = 15 uA 1 = 20 uA 2 = 25 uA 3 = 30 uA 4 = 35 uA 5 = 40 uA 6 = 45 uA 7 = 50 uA
7	Reserved_bit7	RW 0x0	
6:0	REFDIV	RW 0xA	Reference Clock Divider Select Default is set for 40 MHz REFCLK. If a REFCLK frequency other than 40 MHz is used, this value must be changed before powering up the PLL. 0 = bypass, equivalent to divided by 1 1 = illegal 2 = divided by 2 3 = 0x7F = divided by 3 to 127

Table 562: PLL Control 1 Register (n=0–2)
Offset: Port0: 0x000C0004, Port1: 0x000C2004, Port2: 0x000C4004
Offset Formula: 0x000C0004+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:15	Reserved_bit31_15	RW 0x0	Not used
14	PLL_STRESS_TEST	RW 0x0	Stress Test Mode 0 = normal mode 1 = stress test mode
13	CLK_BLK_EN	RW 0x0	480 MHz clock block enable to avoid overshooting issue. 0 = when lock detect and KVCO calibration done, sent the clock out 1 = always sent clock out
12	REFCLK_SEL	RW 0x0	REFCLK select. 0 = To choose REFCLKC_IN1 for USB2 PHY 1 = To choose REFCLKC_IN2

Copyright © 2015 Marvell

Table 562: PLL Control 1 Register (n=0–2) (Continued)
Offset: Port0: 0x000C0004, Port1: 0x000C2004, Port2: 0x000C4004
Offset Formula: 0x000C0004+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
11	CLK160M_EN	RW 0x0	160MHz clock DIG_CLK160M output enable 0 = 160MHz clock is disabled. 1 = 160MHz clock is enabled. It will output REFCLK before PLL is locked, or 160MHz after PLL is locked.
10	TXCLK_EN	RW 0x1	To enable 480MHz clock output. 0 = disabled, no TXCLK 480MHz output 1 = Enabled
9:8	Reserved_bit9_8	RW 0x0	Reserved
7:4	Reserved_bit7_4	RW 0x0	Reserved
3	PLL_SUSPEND_EN	RW 0x1	PLL Suspend enable For multi-port PHY, if PLL_SUSPEND_EN=1 in all channels, uspend=0 will power off PLL; else PLL power will not be affected by uspend. 0 = PLL power is not affected by suspend 1 = SUSPENDM is allowed to power down PLL.
2	PLL_RESET	RW 0x0	PLL Reset 0 = use POR_RESET to reset PLL 1 = reset PLL
1	PU_PLL_BY_REG	RW 0x0	0: Power up PLL by PU_PLL port 1 = by PU_PLL register bit
0	PU_PLL	RW 0x0	Power Up If SUSPENDM from all channels are asserted, and PLL_SUSPEND_EN =1’b1, then PLL will be powered off automatically even PU=1’b1. In addition, PLL is always controlled by pin in scan mode. PLL is always OFF in IDDQ_TEST mode. 0 = power down USB PHY PLL 1 = power up USB PHY PLL

Table 563: Calibration Control Register (n=0–2)
Offset: Port0: 0x000C0008, Port1: 0x000C2008, Port2: 0x000C4008
Offset Formula: 0x000C0008+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31	PLLCAL_DONE	RO 0x0	1: PLL calibration is done 0 = Not done

Copyright © 2015 Marvell

Table 563: Calibration Control Register (n=0–2) (Continued)
Offset: Port0: 0x000C0008, Port1: 0x000C2008, Port2: 0x000C4008
Offset Formula: 0x000C0008+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
30:28	CURRENT_KVCO	RO 0x0	KVCO value currently used
27:24	CURRENT_IMP	RO 0x0	Current IMP value
23	IMPCAL_DONE	RO 0x0	1: Impedance calibration is done 0 = not done
22	PLLCAL_START	RW 0x0	Control KVCO calibration for PLL, every 0->1 transition will start VCO Calibration and keep this bit 1 unless you want to calibrate again. It is better to use VCO calibration to cover all process corners.
21:20	PLLCAL	RW 0x1	Change PLL VCO Calibration Voltage 0 = VDD 1 = 1.0V(default) 2 = 1.05V 3 = 1.1V
19	KVCO_EXT	RW 0x0	Select KVCO Control Signal KVCO_CAL[2:0] 0 = select calibration KVCO 1 = select register control KVCO[2:0].
18:16	KVCO	RW 0x3	PLL KVCO Control Only valid if KVCO_EXT = 1. 0 = high KVCO 7 = Low KVCO
15	VCOCAL_POL	RW 0x0	VCOCAL polarity 0 = Use up/down signal from analog part 1 = Invert up/down signal polarity
14	IMPCAL_POL	RW 0x0	IMPCAL polarity 0 = Use up/down signal from analog part 1 = Invert up/down signal polarity
13	IMPCAL_START	RW 0x0	Control impedance calibration, every 0->1 transition will start impedance Calibration and keep this bit 1 unless you want to calibrate again. It is better to use impedance calibration to cover all process corners.
12	EXT_RCAL_EN	RW 0x0	Impedance Calibration Enable Control 0 = use the auto calibration results 1 = use the value set by EXT_RCAL[3:0]
11	EXT_FS_IMP_EN	RW 0x0	FS Impedance Calibration Enable Control 0 = use the auto calibration results 1 = use the value set by EXT_FS_IMP[3:0]

Copyright © 2015 Marvell

Table 563: Calibration Control Register (n=0–2) (Continued)
Offset: Port0: 0x000C0008, Port1: 0x000C2008, Port2: 0x000C4008
Offset Formula: 0x000C0008+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
10:8	IMPCAL_VTH	RW 0x4	Impedance Calibration Threshold Setting Vth[2:0]: Impedance Each project needs to tune this register respectively. In HS mode, with 45 ohm termination on DP/DM. Sweep register setting to make SEPP equal to 400 mV. 0 = 900/16 = 56 ohms 1 = 900/17 = 53 ohms 4 = 900/20 = 45 ohms 7 = 900/23 = 39 ohms
7:4	EXT_RCAL	RW 0x8	Refer to EXT_FS_IMP[3:0]
3:0	EXT_FS_IMP	RW 0x8	Tx FS driver impedance adjusting for HS loading, target is 45 ohms. Imp_Cal[3:0]: Impedance (SE) This is only used in testing mode, where REG_EX_FS_RCAL_EN = 1. 0 = 900/12 = 75 ohms 1 = 900/13 = 69 ohms 2 = 900/14 = 64 ohms 8 = 900/20 = 45 ohms 15 = VF: 900/27 = 33 ohms

Table 564: TX Channel Control Register 0 (n=0–2)
Offset: Port0: 0x000C000C, Port1: 0x000C200C, Port2: 0x000C400C
Offset Formula: 0x000C000C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:26	Reserved_bit31_26	RW 0x0	Not used
25	PU_BY_REG	RW 0x0	0: TX/RX channel power is controlled by PU_PHYx port The power is always controlled by PU_PHYx pin in scan mode. 1 = the power is controlled by PU_ANA register bit.
24	PU_ANA	RW 0x0	1=power up TX/RX channel analog part 0 = Turn off it
23	PU_VDDR18	RW 0x1	Reserved for analog part
22:20	AMP	RW 0x3	Tx Output Driver Amplitude Control AMP[2 :0] : I (Total) : Amp From ICC current. 0 = 12.6 mA : 283.5 mV 1 = 14.4 mA : 324.0 mV 2 = 16.2 mA : 364.5 mV 3 = 18.0 mA : 405.0 mV 4 = 19.8 mA : 445.5 mV 5 = 21.6 mA : 486.0 mV 6 = 23.4 mA : 526.5 mV 7 = 25.2 mA : 567.0 mV

Copyright © 2015 Marvell

Table 564: TX Channel Control Register 0 (n=0–2) (Continued)
Offset: Port0: 0x000C000C, Port1: 0x000C200C, Port2: 0x000C400C
Offset Formula: 0x000C000C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
19:16	IMP_SEL_LS	RW 0xE	LS TX driver fine adjustment. This field controls 15 cells of the 27 LS TX driver cells. The total number of driver cells enabled is the sum of the cells enabled by DRV_EN_LS and by IMP_SEL_LS. For example, if DRV_EN_LS = 0x3 (enables 6 cells) and IMP_SEL_LS = 0x1 (enables 1 cell), the total number of LS driver cells is 7 IMP_SEL_LS: LS Drive cell control 0 = No cell on 1 = One cell on 2 = Two cells on 15 = VF: 15 cells on
15:12	DRV_EN_LS	RW 0x7	LS TX driver strength coarse control. The whole LS driver includes 27 driver cells. DRV_EN_LS[3:0] controls 12 cells with granularity of 3 cells, IMP_SEL_LS[3:0] control 15 cells with granularity of 1 cell. According to chip test result, around 7 driver cells are optimal for LS eye. It is recommended to change this field to 0x1 by SW after power up. But still need to tune for optimal setting for LS eye with test result. 1 = 3 programmable cell on 3 = 6 programmable cell on 7 = 9 programmable cell on 15 = VF: 12 programmable cell on
11:8	FSDRV_EN	RW 0xF	The whole FS driver includes 12 programmable driver cell and 15 calibration driver cell. 1 = 3 programmable cell on 3 = 6 programmable cell on 7 = 9 programmable cell on 15 = VF: 12 programmable cell on
7	Reserved_bit7	RW 0x0	Reserved
6	LOWVDD_EN	RW 0x1	The design supports low regulator voltage for HS drive, When the regulator voltage for HS drive is low, the driver requires larger device. 0 = For 1.6v 1 = For 1.5v
5:4	HS_SR_SEL	RW 0x2	Controls the rise/fall time of the HS transmitter, 0x00 is shortest rise/fall, 0x11 is longest rise/fall
3:0	HSDRV_EN	RW 0xF	The whole HS driver has been split into four small drivers. The each register of hsdrv_en[3:0] controls one small driver.

Copyright © 2015 Marvell

Table 565: TX Channel Control Register 1 (n=0–2)
Offset: Port0: 0x000C0010, Port1: 0x000C2010, Port2: 0x000C4010
Offset Formula: 0x000C0010+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved_bit31_0	RW 0x0	Reserved

Table 566: Rx Channel Control 0 Register (n=0–2)
Offset: Port0: 0x000C0014, Port1: 0x000C2014, Port2: 0x000C4014
Offset Formula: 0x000C0014+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:30	RX_RESERVE	RW 0x0	Connected to analog module, but reserved for future.
29	CHIRPMODE_SEL	RW 0x0	To select chirp mode 0 is recommended for normal USB operation. 0 = normal chirp mode 1 = special chirp mode
28	SQ_ANA_DTC_SEL	RW 0x0	Analog squelch detect enable 0 = disable analog squelch detect; 1 = enable analog squelch detect
27:26	SQ_ANA_VREF[1:0]	RW 0x0	Analog Squelch Detect reference voltage 0 = DP/DM valley voltage plus 55mv 1 = DP/DM valley voltage plus 60mv 2 = DP/DM valley voltage plus 65mv 3 = DP/DM valley voltage plus 70mv
25	RESET_EXT	RW 0x0	Register Reset signal. 0 = no reset; 1 = reset
24	RESET_EXT_EN	RW 0x0	Reset Signal Select. 0 = select internal reset signal; 1 = select RESET_EXT register
23:20	ReservedTESTMON[3:0]	RW 0x0	Do not modifyControls TEST_PIN analog test point. For Marvell internal use only. Default is high-Z.
19	STRESS_TEST_MODE	RW 0x0	1 : Stress test mode 0 = Normal mode
18:17	Reserved_bit18_17	RW 0x0	Not used
16	SQ_VHL_SEL	RW 0x0	SQ threshold select 0 = Vosl 1 = Vosh

Copyright © 2015 Marvell

Table 566: Rx Channel Control 0 Register (n=0–2) (Continued)
Offset: Port0: 0x000C0014, Port1: 0x000C2014, Port2: 0x000C4014
Offset Formula: 0x000C0014+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
15	SQ_DET_EN	RW 0x1	SQ enable
14	SQ_CM_SEL	RW 0x0	Squelch and Non-Squelch Threshold Common Mode Voltage move down 120 mV-240 mV. When = 1 This should not need to be used. Default 0 should be fine.
13:12	SQ_DLY_SEL	RW 0x1	SQ delay select 0 = no delay 1 = delay 1 clock 2 = delay 2 clocks 3 = delay 3 clocks
11	Reserved_bit11	RW 0x0	Unused
10	LINESTATE_EN	RW 0x1	Linestate enable. Always set to 1.
9:8	DISCON_THRESH	RW 0x2	Host Disconnect Detector Threshold Select Thresh[1:0]: VH (mV) The envelope detector circuits only use half amplitude of the single-ended signal to compare with the threshold 0 = 260 1 = 270 2 = 280 3 = 290
7:6	RXVDDL	RW 0x1	Rx Regulator Output Voltage Control 0 = VDD 1 = 1.1V 2 = 1.2V 3 = 1.3V
5:4	RXVDD18	RW 0x1	1.8V regulator voltage control

Copyright © 2015 Marvell

Table 566: Rx Channel Control 0 Register (n=0–2) (Continued)
Offset: Port0: 0x000C0014, Port1: 0x000C2014, Port2: 0x000C4014
Offset Formula: 0x000C0014+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
3:0	SQ_THRESH	RW 0x7	Squelch and Un-squelch detection threshold setting. Silicon test result shows 0xA works the best. Th[3:0]: Vosh (mV) 0 = 172.5 1 = 172.5 2 = 162.5 3 = 162.5 4 = 152.5 5 = 152.5 6 = 142.5 7 = 142.5 8 = 132.5 10 = VA: 122.5 12 = VC: 112.5 13 = VD: 112.5 14 = VE: 102.5 15 = VF: 102.5VOSH is the nominal squelch threshold. All other columns are for internal information only.

Table 567: Rx Channel Control 1 Register (n=0–2)
Offset: Port0: 0x000C0018, Port1: 0x000C2018, Port2: 0x000C4018
Offset Formula: 0x000C0018+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31	REG_SQCAL_DONE	RO 0x0	Indicate whether squelch calibration done or not 0 = Not done 1 = Done
30:8	Reserved_bit30_8	RW 0x0	Reserved
7:6	SQ_DPDM_AMP_SEL	RW 0x0	SQ calibration threshold select 0 = V130mV: 130mV 1 = V140mV: 140mV 2 = V150mV: 150mV 3 = V160mV: 160mV
5	REG_SQ_UD_SWAP	RW 0x0	Invert analog squelch calibration up/down signal polarity. 0 = Don’t invert 1 = Invert
4	REG_SQCAL_START	RW 0x0	Every rising edge of this register bit starts sq calibration.
3	EXT_SQ_AMP_CAL_EN	RW 0x0	External squelch calibration enable.
2:0	EXT_SQ_AMP_CAL	RW 0x4	External squelch calibration number

Copyright © 2015 Marvell

Table 568: Digital Control Register 0 (n=0–2)
Offset: Port0: 0x000C001C, Port1: 0x000C201C, Port2: 0x000C401C
Offset Formula: 0x000C001C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31	BITSTUFFING_ERROR	RW 0x0	HS bit stuffing error flag. Write 1 to clear.
30	LOSS_OF_SYNC_ERROR	RW 0x0	HS Loss of SYNC error flag. Write 1 to clear.
29	FIFO_OVF_ERROR	RW 0x0	HS Rx FIFO Overflow error flag. Write 1 to clear.
28:27	Reserved_bit28_27	RW 0x0	Unused
26	CORE_UTMI_SEL	RW 0x0	UTMI signal select 0 = Allow reg_lat_utmi to work, actual UTMI signals are not updated during packet is transferring 1 = Always use UTMI signals from controller. This is the same as previous version.
25	SE1_FILT_SEL	RW 0x1	Linestate SE1 filter select 0 = To filter SE1 1 = don't filter
24	SE0_FILT_SEL	RW 0x1	Linestate SE0 filter select 0 = To filter SE0 1 = don't filter
23	EDGE_OPT_EN	RW 0x1	To select if optimize edge 0 = No edge optimization 1 = Optimize edge
22	STOP_AT_RXERR	RW 0x0	To select to stop Rx when RX_ERROR is detected or not. 0 = RX_ACTIVE will last until SQ = 1 1 = RX_ACTIVE is de-asserted after Rx error is found. The EARLY_TX_ENABLE bit must be set to 0.
21	RX_RUNAWAY_EN	RW 0x0	Disables Rx run away in case EOP is not detected. 0 = Rx run away is eliminated 1 = Rx run away is allowed
20	LOSSOFSYNC_EN	RW 0x1	Enables loss of SYNC error to propagate to LOSS_OF_SYNC_ERROR register. 0 = disable 1 = enable
19	FIFOOVF_EN	RW 0x1	Enables FIFO overflow error to propagate to RX_ERROR. 0 = FIFO_OVF will not assert RX_ERROR 1 = enable

Copyright © 2015 Marvell

Table 568: Digital Control Register 0 (n=0–2) (Continued)
Offset: Port0: 0x000C001C, Port1: 0x000C201C, Port2: 0x000C401C
Offset Formula: 0x000C001C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
18:16	SQ_FILT	RW 0x0	SQ Glitch Filtering Width Select. This eliminates the glitch when SQ = 0. 0 = no filtering 1 = filter two bits glitch 2 = filter four bits glitch 3 = filter six bits glitch 4 = filter eight bits glitch 5 = filter 10 bits glitch 6 = filter 12 bits glitch 7 = filter 14 bits glitch
15	Reserved_bit15	RW 0x1	Unused
14:12	SQ_BLK	RW 0x0	To set the HS data bits to block after SQ falling edge. The blocked data bits are ignored. 0 = no blocking 1 = block one bits 2 = block two bits 3 = block three bits 4 = block four bits 5 = block five bits 6 = block six bits 7 = block seven bits
11	EARLY_TX_ENABLE	RW 0x0	Early Tx Enable 0 = after controller asserts TX_VALID, PHY will check USB bus squelch status. If it is squelch, Tx will start immediately. Else it will wait for squelch. 1 = HS Tx will start after controller asserts TX_VALID
10	RXFILT2_EN	RW 0x1	0 = no filter2 Data 001100 is corrected to 000000. Data 110011 is corrected to 111111. 1 = enable Rx data sample filter2
9	RXFILT1_EN	RW 0x1	0 = no filter1 Data 010 is corrected to 000. Data 101 is corrected to 111. 1 = enable Rx data sample filter1
8	LONG_EOP	RW 0x0	Compatible to old controller, HS EOP is 40 bits long.
7	PLL_LOCK_BYPASS	RW 0x0	PLL Lock Bypass 0 = output CLK_OUT after PLL is locked 1 = output CLK_OUT after PLL is on
6	CLK_SUSPEND_EN	RW 0x1	Clock Suspend 0 = CLK_OUT will be free-running clock 1 = enable UTMI_SUSPENDM to stop CLK_OUT

Copyright © 2015 Marvell

Table 568: Digital Control Register 0 (n=0–2) (Continued)
Offset: Port0: 0x000C001C, Port1: 0x000C201C, Port2: 0x000C401C
Offset Formula: 0x000C001C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
5	HOST_DISCON_SEL1	RW 0x1	Host Disconnect Select 1 0 = enable uspend to gate off HOST_DISCONNECT 1 = uspend does not gate off HOST_DISCONNECT
4	HOST_DISCON_SEL0	RW 0x0	Host Disconnect Select 0 0 = sample HS_DISCONNECT at the end of SOF EOP only 1 = sample HS_DISCONNECT during the whole SOF EOP
3	SYNCDET_WINDOW_EN	RW 0x1	Defines a time window after non-squelch as SYNC detection period. 0 = detect SYNC as long as squelch = 0 1 = detect SYNC in the first 64 or 40 HS bits time after squelch = 0
2	SYNCDET_WIN32	RW 0x1	SYNC Detect Window Select 0 = 64-bit window after SQ = 0 1 = 40-bit window after SQ = 0
1:0	SYNC_NUM	RW 0x1	Minimum SYNC Detect Number of Bits 0 = 00 1 = 0010 2 = 001010 3 = 00101010

Table 569: Digital Control Register 1 (n=0–2)
Offset: Port0: 0x000C0020, Port1: 0x000C2020, Port2: 0x000C4020
Offset Formula: 0x000C0020+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31	Reserved_bit31	RW 0x0	Not used
30	REG_LSIDLE_PRE	RW 0x0	0: To drive FS idle after PRE is sent 1 = To drive LS idle after PRE is sent
29:22	REG_CLK_OUT_DLY	RW 0x8	To set CLK_OUT delay after analog part is powered up or SUSPENDM rising edge. CLK_OUT will start to output XCVR_CLK after delay defined by this register. 0 = no delay 1 = delay by 256 HS bit time 2 = 512 HS bits 8 = about 4uS 48 = about 25 uS 96 = about 51 uS 144 = about 76 uS 188 = VBC: about 100 us 255 = VFF: about 136 us

Copyright © 2015 Marvell

Table 569: Digital Control Register 1 (n=0–2) (Continued)
Offset: Port0: 0x000C0020, Port1: 0x000C2020, Port2: 0x000C4020
Offset Formula: 0x000C0020+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
21	REG_SAMPLER_PREOFF	RW 0x1	When REG_SAMPLER_ON=0, this bit controls if the sampler is turned off 2 HS bits before TX_ENABLE=1 0 = Sampler is OFF at TX_ENABLE rising edge 1 = Sampler is OFF 2 HS bits before TX_ENABLE
20	REG_SAMPLER_ON	RW 0x0	To control if the sampler is always ON or under register control. 0 = The RX sampler is controlled by REG_SAMPLER_PREOFF and REG_SAMPLEROFF_DLY [3:0] 1 = Force sampler always ON
19:16	REG_SAMPLEROFF_DLY	RW 0x8	When REG_SAMPLER_ON =0, this register controls the bits the sampler is turned off after TX_ENABLE = 0. Each step controls two HS bits OFF. 0 = sampler is turned on at Tx_enable falling edge 1 = sampler is turned on 2 HS bits after Tx_enable falling edge 8 = sampler is turned on 16 HS bits after Tx_enable falling edge 15 = VF: sampler is turned on 30 HS bits after Tx_enable falling edge
15	REG_SWAP_DPDM	RW 0x0	To control if swap DP/DM pad. 0 = DP/DM pads work as USB D+, D- 1 = DP/DM pads work as USB D-, D+
14	REG_LAT_UTMI	RW 0x0	To latch UTMI XCVR_SELECT, OPMODE and TERM_SELECT signal 0 = Do not latch 1 = latch except during suspend
13	REG_OPMODE_SEL	RW 0x0	OPMODE select 0 = To use opmode from USB controller directly 1 = to use opmode at tx_valid rising edge.
12	REG_FLSMODELP_EN	RW 0x1	FS/LS serial mode low power enable 0 = Do not enable FS/LS serial mode low power 1 = enable
11	REG_FREERUNCLK_EN	RW 0x0	Static enable for free running clock output 0 = free running clock is disabled 1 = free running clock is enabled
10	REG_NOVBUS_DPDM00	RW 0x1	When VBUS is off: When VBUS is OFF, DP/DM is pulled low by default. 0 = does not pull-down DP/DM pads 1 = pull-down DP/DM pads to 0
9:8	Reserved_bit9_8	RW 0x0	
7	REG_FS_EOP_MODE	RW 0x1	Enable 1-3 SE0 FS EOP Detection

Copyright © 2015 Marvell

Table 569: Digital Control Register 1 (n=0–2) (Continued)
Offset: Port0: 0x000C0020, Port1: 0x000C2020, Port2: 0x000C4020
Offset Formula: 0x000C0020+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
6	REG_DP_PULLDOWN	RW 0x0	Pull-Down DP 0 = no DP pull-down.(default) 1 = manual DP pull-down
5	REG_DM_PULLDOWN	RW 0x0	Pull-Down DM 0 = no DM pull-down.(default) 1 = manual DP pull-down
4	REG_ARC_DPDM_MODE	RW 0x1	Use dp_pulldown and dm_pulldown from USB controller. 0 = use register programmed pull-down 1 = user USB controller pull-down (default)
3	REG_FS_RX_ERROR_MODE2	RW 0x1	Disable multiple FS_RX_ERROR within a packet
2	REG_FS_RX_ERROR_MODE1	RW 0x1	Enable fast FS_RX_ERROR return to 0. Valid only with REG_RX_ERROR_MODE = 1.
1	REG_FS_RX_ERROR_MODE	RW 0x1	Enable FS RX_ERROR return to 0 at 1 FS bit time
0	REG_FORCE_END_EN	RW 0x1	Enable force RX_ACTIVE to 0 when there are consecutive bit-stuff errors with length over 14FS bits time

Table 570: Test Control and Status Register 0 (n=0–2)
Offset: Port0: 0x000C0024, Port1: 0x000C2024, Port2: 0x000C4024
Offset Formula: 0x000C0024+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31	TEST_FAIL	RO 0x0	Test Error Flag This flag must be read after TEST_DONE changes from 0 to 1. Otherwise the read value is undefined. 0 = error not occurred 1 = error occurred
30	TEST_DONE	RO 0x0	Test Finish Indicator If TEST_DONE is not asserted after the time it should have been set, it is also a failed chip. 0 = test not finished 1 = test finished
29	REG_TEST_EN	RW 0x0	Test Enable (1=Start loopback test)
28	REG_TEST_DIG_LPBK	RW 0x0	Enable Digital Loopback 0 = analog loopback 1 = digital loopback

Copyright © 2015 Marvell

Table 570: Test Control and Status Register 0 (n=0–2) (Continued)
Offset: Port0: 0x000C0024, Port1: 0x000C2024, Port2: 0x000C4024
Offset Formula: 0x000C0024+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
27	REG_TEST_ANA_LPBK	RW 0x0	Enable analog loopback test mode 0 = analog loopback not enabled 1 = analog loopback enabled
26	REG_TEST_BYPASS	RW 0x0	Enable Bypass Mode (Bypass bit-stuff/encoder)
25	REG_TEST_UTMI_SEL	RW 0x0	Enable UTMI Test Registers: TEST_TX_BITSTUFF_EN, TEST_SUSPENDM, TEST_RESET, TEST_TERM_SELECT, TEST_XCVR_SELECT, TEST_OPMODE. This bit must be set to 0 for normal operation mode.
24	Reserved_bit24	RW 0x0	Not used
23	REG_TEST_TX_BITSTUFF_EN	RW 0x1	UTMI Test Mode tx_bitstuff_en
22	REG_TEST_SUSPENDM	RW 0x1	UTMI Test Mode suspendm
21	REG_TEST_RESET	RW 0x0	UTMI Test Mode Reset
20	REG_TEST_TERM_SELECT	RW 0x0	UTMI Test Mode term_select
19:18	REG_TEST_XVCR_SELECT	RW 0x0	UTMI Test Mode xcvr_select[1:0]
17:16	REG_TEST_OP_MODE	RW 0x0	UTMI Test Mode op_mode
15	Reserved_bit15	RW 0x0	Unused
14:12	REG_TEST_SKIP	RW 0x0	Digital Loopback Starting Position 0 = no rotation 1 = rotate by 1-bit 2 = rotate by 2-bit 3 = rotate by 3-bit 4 = rotate by 4-bit 5 = rotate by 5-bit 6 = rotate by 6-bit 7 = rotate by 7-bit
11:10	Reserved_bit11_10	RW 0x0	Unused
9:8	REG_TEST_MODE	RW 0x0	Programmable Predefined Tx Pattern 0 = use TX_pattern register 1 = incremental pattern. 00, 01, 02, ..., FF, 00 2 = circular pattern = 01->02->04->08->10->20->40->80->01 3 = prbs pattern

Copyright © 2015 Marvell

Table 570: Test Control and Status Register 0 (n=0–2) (Continued)
Offset: Port0: 0x000C0024, Port1: 0x000C2024, Port2: 0x000C4024
Offset Formula: 0x000C0024+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
7:0	REG_TEST_TX_PATTERN	RW 0x0	Programmable Tx Pattern For 16-bit mode, Bits[15:8] pattern is the same as Bits[7:0]. This pattern register is used when TEST_MODE[1:0]=2'b00.

Table 571: Test Control and Status Register 1 (n=0–2)
Offset: Port0: 0x000C0028, Port1: 0x000C2028, Port2: 0x000C4028
Offset Formula: 0x000C0028+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:11	Reserved_bit31_11	RW 0x0	Reserved
10:0	TEST_LEN	RW 0x0	Loopback test length. This is a new register for V2.0. The loopback test length can be any number of bytes from 1 to 2047. It is recommended not to run loopback test longer than 1024 bytes. 0 = non-stop loopback test 1 = 0x7FF: 1 byte to 2047 bytes.

Table 572: Monitor Register (n=0–2)
Offset: Port0: 0x000C002C, Port1: 0x000C202C, Port2: 0x000C402C
Offset Formula: 0x000C002C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved_bit31_16	RO 0x0	ReservedUSB2 PHY MON_BUS read out. For Marvell internal use only.
15:0	Reserved_bit15_0	RO 0x0	ReservedPHY_MON digital Test Bus Select For Marvell internal use only. Only bit [20:16] are used. Other bits are reserved.

Table 573: PHY RESERVE Register (n=0–2)
Offset: Port0: 0x000C0030, Port1: 0x000C2030, Port2: 0x000C4030
Offset Formula: 0x000C0030+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved_bit31_0	RW 0x0	Reserved

Copyright © 2015 Marvell

Table 574: PHY Charger Detector Control Register (n=0–2)
Offset: Port0: 0x000C0038, Port1: 0x000C2038, Port2: 0x000C4038
Offset Formula: 0x000C0038+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:14	Reserved_bit31_14	RW 0x0	Reserved
13	ENABLE_SWITCH_DM	RW 0x0	Control connection between the charge detect circuit and DM port. 0 = Turn Off the Switch 1 = Turn On the Switch
12	ENABLE_SWITCH_DP	RW 0x0	Control connection between the charge detect circuit and DP port. 0 = Turn Off the Switch 1 = Turn On the Switch
11:10	VSRC_CHARGE	RW 0x0	Controlled VSRC Reference Voltage 0 = 0.55V 1 = 0.6V(Default) 2 = 0.65V 3 = 0.7V
9:8	VDAT_CHARGE	RW 0x0	Controlled VDAT Reference Voltage 0 = 0.3V 1 = 0.325V(Default) 2 = 0.35V 3 = 0.4V
7	CDP_DM_AUTO_SWITCH	RW 0x0	1: Automatic turn on DM(D_MINUS) port switch when CHG_DTC_OUT=1. This is for CDP Detection only. 0 = DM Switch still controlled by ENABLE_SWITCH_DM
6	DP_DM_SWAP_CTRL	RW 0x0	This bit is used for the charger detector. 0 = For portable device use. D+ is driven to 0.6V and D- is used to detect charger present or not. 1 = for down stream port charging. D- is driven to 0.6V and D+ is used to detect charger present or not
5	PU_CHRG_DTC	RW 0x0	Power-up signal for the charge detect circuit. It is forced to 0 in scan mode or IDDQ test mode It is forced OFF in scan mode 0 = Power off 1 = Power on
4	PD_EN	RW 0x0	1: To turn on portable device mode It is forced OFF in scan mode 0 = Turn off
3	DCP_EN	RW 0x0	1: To turn on dedicated charger port mode It is forced OFF in scan mode 0 = Turn off
2	CDP_EN	RW 0x0	1: To turn on charging downstream port mode It is forced OFF in scan mode 0 = Turn off

Copyright © 2015 Marvell

Table 574: PHY Charger Detector Control Register (n=0–2) (Continued)
Offset: Port0: 0x000C0038, Port1: 0x000C2038, Port2: 0x000C4038
Offset Formula: 0x000C0038+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
1:0	TESTMON_CHRGDTC	RW 0x0	Analog DC test control signal for charge detector. It is forced to 2'b00 in scan mode It is forced to2'b00 in scan mode 0 = tri-state at TEST_PIN 1 = VREF_0P6V 0.6V 2 = VDAT_REF 0.325V 3 = CHRG_DTC_OUT 0 or Dvdd

Table 575: Reserved. Register (n=0–2)
Offset: Port0: 0x000C003C, Port1: 0x000C203C, Port2: 0x000C403C
Offset Formula: 0x000C003C+0x2000*n: where n (0-2) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved_bit31_0	RW 0x0	Reserved

Copyright © 2015 Marvell

A.11 USB 3.0 Host Controller Registers

NOTE: For more information about the USB 3.0 Host Controller registers, refer to the xHCI Specification, Revision 1.0.

The following table provides a summarized list of all registers that belong to the USB 3.0 Host Controller, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 576: Summary Map Table for the USB 3.0 Host Controller Registers

Register Name	Offset	Table and Page
USB 3.0 Host		
USB 3.0 Host Controller Capability		
CAPLENGTH Register (n=0–1)	usb host port0: 0x000F0000, usb host port1: 0x000F8000	Table 577, p. 720
HCIVERSION Register (n=0–1)	usb host port0: 0x000F0002, usb host port1: 0x000F8002	Table 578, p. 720
HCSPARAMS_1 Register (n=0–1)	usb host port0: 0x000F0004, usb host port1: 0x000F8004	Table 579, p. 721
HCSPARAMS_2 Register (n=0–1)	usb host port0: 0x000F0008, usb host port1: 0x000F8008	Table 580, p. 721
HCSPARAMS_3 Register (n=0–1)	usb host port0: 0x000F000C, usb host port1: 0x000F800C	Table 581, p. 722
HCCPARAMS Register (n=0–1)	usb host port0: 0x000F0010, usb host port1: 0x000F8010	Table 582, p. 722
DBOFF Register (n=0–1)	usb host port0: 0x000F0014, usb host port1: 0x000F8014	Table 583, p. 723
RTSOFF Register (n=0–1)	usb host port0: 0x000F0018, usb host port1: 0x000F8018	Table 584, p. 723
USB 3.0 Host Controller Operational		
USBCMD Register (n=0–1)	usb host port0: 0x000F0020, usb host port1: 0x000F8020	Table 585, p. 724
USBSTS Register (n=0–1)	usb host port0: 0x000F0024, usb host port1: 0x000F8024	Table 586, p. 724
PAGESIZE Register (n=0–1)	usb host port0: 0x000F0028, usb host port1: 0x000F8028	Table 587, p. 725
DNCTRL Register (n=0–1)	usb host port0: 0x000F0034, usb host port1: 0x000F8034	Table 588, p. 725
CRCR Register (n=0–1)	usb host port0: 0x000F0038, usb host port1: 0x000F8038	Table 589, p. 726
DCBAAP Register (n=0–1)	usb host port0: 0x000F0050, usb host port1: 0x000F8050	Table 590, p. 727
xHCI_CONFIG Register (n=0–1)	usb host port0: 0x000F0058, usb host port1: 0x000F8058	Table 591, p. 727
USB3_PORTSC Register (n=0–1)	usb host port0: 0x000F0420, usb host port1: 0x000F8420	Table 592, p. 727
USB3_PORTPMSC Register (n=0–1)	usb host port0: 0x000F0424, usb host port1: 0x000F8424	Table 593, p. 729

Copyright © 2015 Marvell

Table 576: Summary Map Table for the USB 3.0 Host Controller Registers (Continued)

Register Name	Offset	Table and Page
USB3_PORTLI Register (n=0–1)	usb host port0: 0x000F0428, usb host port1: 0x000F8428	Table 594, p. 729
USB3_PORTHLPMC Register (n=0–1)	usb host port0: 0x000F042C, usb host port1: 0x000F842C	Table 595, p. 729
USB2_PORTSC Register (n=0–1)	usb host port0: 0x000F0430, usb host port1: 0x000F8430	Table 596, p. 730
USB2_PORTPMSC Register (n=0–1)	usb host port0: 0x000F0434, usb host port1: 0x000F8434	Table 597, p. 731
USB_PORTLI Register (n=0–1)	usb host port0: 0x000F0438, usb host port1: 0x000F8438	Table 598, p. 731
USB2_PORTHLPMC Register (n=0–1)	usb host port0: 0x000F043C, usb host port1: 0x000F843C	Table 599, p. 732
USB 3.0 Host Controller Runtime		
MFINDEX Register (n=0–1)	usb host port0: 0x000F1000, usb host port1: 0x000F9000	Table 600, p. 732
IMAN Register (n=0–1)	usb host port0: 0x000F1020, usb host port1: 0x000F9020	Table 601, p. 732
IMOD Register (n=0–1)	usb host port0: 0x000F1024, usb host port1: 0x000F9024	Table 602, p. 733
ERSTSZ Register (n=0–1)	usb host port0: 0x000F1028, usb host port1: 0x000F9028	Table 603, p. 733
ERSTBA Register (n=0–1)	usb host port0: 0x000F1030, usb host port1: 0x000F9030	Table 604, p. 733
ERDP Register (n=0–1)	usb host port0: 0x000F1038, usb host port1: 0x000F9038	Table 605, p. 733
USB 3.0 Host Controller Doorbell		
CMD_DB Register (n=0–1)	usb host port0: 0x000F2000, usb host port1: 0x000FA000	Table 606, p. 734
DB1 Register (n=0–1)	usb host port0: 0x000F2004, usb host port1: 0x000FA004	Table 607, p. 734
DB2 Register (n=0–1)	usb host port0: 0x000F2008, usb host port1: 0x000FA008	Table 608, p. 734
DB3 Register (n=0–1)	usb host port0: 0x000F200C, usb host port1: 0x000FA00C	Table 609, p. 735
DB4 Register (n=0–1)	usb host port0: 0x000F2010, usb host port1: 0x000FA010	Table 610, p. 735
DB5 Register (n=0–1)	usb host port0: 0x000F2014, usb host port1: 0x000FA014	Table 611, p. 735
DB6 Register (n=0–1)	usb host port0: 0x000F2018, usb host port1: 0x000FA018	Table 612, p. 735
DB7 Register (n=0–1)	usb host port0: 0x000F201C, usb host port1: 0x000FA01C	Table 613, p. 736
DB8 Register (n=0–1)	usb host port0: 0x000F2020, usb host port1: 0x000FA020	Table 614, p. 736
DB9 Register (n=0–1)	usb host port0: 0x000F2024, usb host port1: 0x000FA024	Table 615, p. 736
DB10 Register (n=0–1)	usb host port0: 0x000F2028, usb host port1: 0x000FA028	Table 616, p. 736

Document Classification: Proprietary Information

Table 576: Summary Map Table for the USB 3.0 Host Controller Registers (Continued)

Register Name	Offset	Table and Page
DB11 Register (n=0–1)	usb host port0: 0x000F202C, usb host port1: 0x000FA02C	Table 617, p. 737
DB12 Register (n=0–1)	usb host port0: 0x000F2030, usb host port1: 0x000FA030	Table 618, p. 737
DB13 Register (n=0–1)	usb host port0: 0x000F2034, usb host port1: 0x000FA034	Table 619, p. 737
DB14 Register (n=0–1)	usb host port0: 0x000F2038, usb host port1: 0x000FA038	Table 620, p. 737
DB15 Register (n=0–1)	usb host port0: 0x000F203C, usb host port1: 0x000FA03C	Table 621, p. 738
DB16 Register (n=0–1)	usb host port0: 0x000F2040, usb host port1: 0x000FA040	Table 622, p. 738
DB17 Register (n=0–1)	usb host port0: 0x000F2044, usb host port1: 0x000FA044	Table 623, p. 738
DB18 Register (n=0–1)	usb host port0: 0x000F2048, usb host port1: 0x000FA048	Table 624, p. 738
DB19 Register (n=0–1)	usb host port0: 0x000F204C, usb host port1: 0x000FA04C	Table 625, p. 739
DB20 Register (n=0–1)	usb host port0: 0x000F2050, usb host port1: 0x000FA050	Table 626, p. 739
DB21 Register (n=0–1)	usb host port0: 0x000F2054, usb host port1: 0x000FA054	Table 627, p. 739
DB22 Register (n=0–1)	usb host port0: 0x000F2058, usb host port1: 0x000FA058	Table 628, p. 739
DB23 Register (n=0–1)	usb host port0: 0x000F205C, usb host port1: 0x000FA05C	Table 629, p. 740
DB24 Register (n=0–1)	usb host port0: 0x000F2060, usb host port1: 0x000FA060	Table 630, p. 740
DB25 Register (n=0–1)	usb host port0: 0x000F2064, usb host port1: 0x000FA064	Table 631, p. 740
DB26 Register (n=0–1)	usb host port0: 0x000F2068, usb host port1: 0x000FA068	Table 632, p. 740
DB27 Register (n=0–1)	usb host port0: 0x000F206C, usb host port1: 0x000FA06C	Table 633, p. 741
DB28 Register (n=0–1)	usb host port0: 0x000F2070, usb host port1: 0x000FA070	Table 634, p. 741
DB29 Register (n=0–1)	usb host port0: 0x000F2074, usb host port1: 0x000FA074	Table 635, p. 741
DB30 Register (n=0–1)	usb host port0: 0x000F2078, usb host port1: 0x000FA078	Table 636, p. 741
DB31 Register (n=0–1)	usb host port0: 0x000F207C, usb host port1: 0x000FA07C	Table 637, p. 742
USB 3.0 Host Controller Extended Capability		
xECP_SUP_CAP_USB2_R1 Register (n=0–1)	usb host port0: 0x000F2800, usb host port1: 0x000FA800	Table 638, p. 742
xECP_SUP_CAP_USB2_R2 Register (n=0–1)	usb host port0: 0x000F2804, usb host port1: 0x000FA804	Table 639, p. 742

Table 576: Summary Map Table for the USB 3.0 Host Controller Registers (Continued)

Register Name	Offset	Table and Page
xECP_SUP_CAP_USB2_R3 Register (n=0–1)	usb host port0: 0x000F2808, usb host port1: 0x000FA808	Table 640, p. 742
xECP_SUP_CAP_USB3_R1 Register (n=0–1)	usb host port0: 0x000F2810, usb host port1: 0x000FA810	Table 641, p. 743
xECP_SUP_CAP_USB3_R2 Register (n=0–1)	usb host port0: 0x000F2814, usb host port1: 0x000FA814	Table 642, p. 743
xECP_SUP_CAP_USB3_R3 Register (n=0–1)	usb host port0: 0x000F2818, usb host port1: 0x000FA818	Table 643, p. 744

A.11.1 USB 3.0 Host

A.11.1.1 USB 3.0 Host Controller Capability

Table 577: CAPLENGTH Register (n=0–1)
Offset: usb host port0: 0x000F0000, usb host port1: 0x000F8000
Offset Formula: 0x000F0000+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.1 xHCI spec Rev1.0			
15:8	Rserved	RO 0x0	Rserved
7:0	CAPLENGTH	RO 0x20	CAPLENGTH

Table 578: HCIVERSION Register (n=0–1)
Offset: usb host port0: 0x000F0002, usb host port1: 0x000F8002
Offset Formula: 0x000F0002+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.2 xHCI spec Rev1.0			
31:16	Reserved	RSVD 0x0	Reserved
15:0	HCIVERSION	RO 0x100	HCIVERSION

Document Classification: Proprietary Information

Table 579: HCSPARAMS_1 Register (n=0–1)
Offset: usb host port0: 0x000F0004, usb host port1: 0x000F8004
Offset Formula: 0x000F0004+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.3 xHCI spec Rev1.0			
31:24	MaxPorts	RO 0x2	MaxPorts
23:19	Reserved	RO 0x0	Reserved
18:8	MaxIntrs	RO 0x1	MaxIntrs
7:0	MaxSlots	RO 0x20	MaxSlots

Table 580: HCSPARAMS_2 Register (n=0–1)
Offset: usb host port0: 0x000F0008, usb host port1: 0x000F8008
Offset Formula: 0x000F0008+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.4 xHCI spec Rev1.0			
31:27	MaxSPB_Lo	RO 0x1	MaxSPB_Lo
26	SPR	RO 0x1	SPR
25:21	MAX_SPB_HI	RO 0x0	MAX_SPB_HI
20:8	Reserved	RO 0x0	Reserved
7:4	ERST_MAX	RO 0xF	ERST_MAX
3:0	IST	RO 0x2	IST

Copyright © 2015 Marvell

Table 581: HCSPARAMS_3 Register (n=0–1)
Offset: usb host port0: 0x000F000C, usb host port1: 0x000F800C
Offset Formula: 0x000F000C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.5 xHCI spec Rev1.0			
31:16	U2_DEV_EXIT_LATENCY	RO 0x7FF	U2_DEV_EXIT_LATENCY Worst case latency to transition from U2 to U0. Applies to all hub ports. Valid: 0x0000 - 0x07FF Reserved: 0x0800 - 0xFFFF
15:8	Reserved	RO 0x0	Reserved
7:0	U1_DEV_EXIT_LATENCY	RO 0xA	U1_DEV_EXIT_LATENCY Worst case latency to transition from U1 to U0. Applies to all hub ports. Valid: 0x00 - 0x0A Reserved: 0x0B - 0xFF

Table 582: HCCPARAMS Register (n=0–1)
Offset: usb host port0: 0x000F0010, usb host port1: 0x000F8010
Offset Formula: 0x000F0010+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.6 xHCI spec Rev1.0			
31:16	xECP	RO 0xA00	xECP
15:12	MaxPSASize	RO 0x0	MaxPSASize
11	CFC	RO 0x0	CFC
10	SEC	RO 0x0	SEC
9	SPC	RO 0x0	SPC
8	PAE	RO 0x1	PAE
7	NSS	RO 0x1	NSS
6	LTC	RO 0x0	LTC
5	LHRC	RO 0x0	LHRC

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 582: HCCPARAMS Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0010, usb host port1: 0x000F8010
Offset Formula: 0x000F0010+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
4	PIND	RO 0x1	PIND
3	PPC	RO 0x1	PPC
2	CSZ	RO 0x0	CSZ
1	BNC	RO 0x0	BNC
0	AC64	RO 0x0	AC64

Table 583: DBOFF Register (n=0–1)
Offset: usb host port0: 0x000F0014, usb host port1: 0x000F8014
Offset Formula: 0x000F0014+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.7 xHCI spec Rev1.0			
31:2	DBOFF	RO 0x800	DBOFF
1:0	Reserved	RO 0x0	Reserved

Table 584: RTSOFF Register (n=0–1)
Offset: usb host port0: 0x000F0018, usb host port1: 0x000F8018
Offset Formula: 0x000F0018+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.3.8 xHCI spec Rev1.0			
31:5	RTSOFF	RO 0x80	RTSOFF
4:0	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

A.11.1.2 USB 3.0 Host Controller Operational

Table 585: USBCMD Register (n=0–1)
Offset: usb host port0: 0x000F0020, usb host port1: 0x000F8020
Offset Formula: 0x000F0020+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.2 xHCI spec Rev1.0			
31:12	Reserved	RO 0x0	Reserved
11	EU3S	RW 0x0	EU3S
10	EWE	RW 0x0	EWE
9	CRS	RW 0x0	CRS
8	CSS	RW 0x0	CSS
7	LHCRST	RO 0x0	LHCRST
6:4	Reserved	RO 0x0	Reserved
3	HSEE	RW 0x0	HSEE
2	INTE	RW 0x0	INTE
1	HCRST	RW 0x0	HCRST
0	Run_Stop	RW 0x0	Run_Stop

Table 586: USBSTS Register (n=0–1)
Offset: usb host port0: 0x000F0024, usb host port1: 0x000F8024
Offset Formula: 0x000F0024+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.1 xHCI spec Rev1.0			
31:13	Reserved	RO 0x0	Reserved
12	HCE	RO 0x0	HCE
11	CNR	RO 0x1	CNR
10	SRE	RW 0x0	SRE

Document Classification: Proprietary Information

Table 586: USBSTS Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0024, usb host port1: 0x000F8024
Offset Formula: 0x000F0024+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
9	RSS	RO 0x0	RSS
8	SSS	RO 0x0	SSS
7:5	Reserved	RO 0x0	Reserved
4	PCD	RW 0x0	PCD
3	EINT	RW 0x0	EINT
2	HSE	RW 0x0	HSE
1	Reserved	RO 0x0	Reserved
0	HCH	RO 0x1	HCH

Table 587: PAGESIZE Register (n=0–1)
Offset: usb host port0: 0x000F0028, usb host port1: 0x000F8028
Offset Formula: 0x000F0028+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.3 xHCI spec Rev1.0			
31:16	Reserved	RO 0x0	Reserved
15:0	PAGESIZE	RO 0x1	PAGESIZE

Table 588: DNCTRL Register (n=0–1)
Offset: usb host port0: 0x000F0034, usb host port1: 0x000F8034
Offset Formula: 0x000F0034+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.4 xHCI spec Rev1.0			
31:16	Reserved	RO 0x0	Reserved
15	N15	RW 0x0	N15
14	N14	RW 0x0	N14

Copyright © 2015 Marvell

Table 588: DNCTRL Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0034, usb host port1: 0x000F8034
Offset Formula: 0x000F0034+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
13	N13	RW 0x0	N13
12	N12	RW 0x0	N12
11	N11	RW 0x0	N11
10	N10	RW 0x0	N10
9	N9	RW 0x0	N9
8	N8	RW 0x0	N8
7	N7	RW 0x0	N7
6	N6	RW 0x0	N6
5	N5	RW 0x0	N5
4	N4	RW 0x0	N4
3	N3	RW 0x0	N3
2	N2	RW 0x0	N2
1	N1	RW 0x0	N1
0	N0	RW 0x0	N0

Table 589: CRCR Register (n=0–1)
Offset: usb host port0: 0x000F0038, usb host port1: 0x000F8038
Offset Formula: 0x000F0038+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.5 xHCI spec Rev1.0			
31:6	CRP_Lo	RW 0x0	CRP_Lo
5:4	Reserved	RO 0x0	Reserved
3	CRR	RO 0x0	CRR

Copyright © 2015 Marvell

Table 589: CRCR Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0038, usb host port1: 0x000F8038
Offset Formula: 0x000F0038+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
2	CA	RW 0x0	CA
1	CS	RW 0x0	CS
0	RCS	RW 0x0	RCS

Table 590: DCBAAP Register (n=0–1)
Offset: usb host port0: 0x000F0050, usb host port1: 0x000F8050
Offset Formula: 0x000F0050+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.6 xHCI spec Rev1.0			
31:6	DCBAAP_Lo	RW 0x0	DCBAAP_Lo
5:0	Reserved	RO 0x0	Reserved

Table 591: xHCI_CONFIG Register (n=0–1)
Offset: usb host port0: 0x000F0058, usb host port1: 0x000F8058
Offset Formula: 0x000F0058+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.7 xHCI spec Rev1.0			
31:8	Reserved	RO 0x0	Reserved
7:0	MaxSlotsEn	RW 0x0	MaxSlotsEn

Table 592: USB3_PORTSC Register (n=0–1)
Offset: usb host port0: 0x000F0420, usb host port1: 0x000F8420
Offset Formula: 0x000F0420+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.8 xHCI spec Rev1.0			
31	WPR	RW 0x0	WPR
30	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 592: USB3_PORTSC Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0420, usb host port1: 0x000F8420
Offset Formula: 0x000F0420+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
29:28	Reserved	RO 0x0	Reserved
27	WOE	RW 0x0	WOE
26	WDE	RW 0x0	WDE
25	WCE	RW 0x0	WCE
24	Reserved	RO 0x0	Reserved
23	CEC	RW 0x0	CEC
22	PLC	RW 0x0	PLC
21	PRC	RW 0x0	PRC
20	Reserved	RO 0x0	Reserved
19	WRC	RW 0x0	WRC
18	PEC	RW 0x0	PEC
17	CSC	RW 0x0	CSC
16	LWS	RW 0x0	LWS
15:14	PIC	RW 0x0	PIC
13:10	Port_Speed	RO 0x0	Port_Speed
9	PP	RW 0x1	PP
8:5	PLS	RW 0x5	PLS
4	PR	RW 0x0	PR
3	Reserved	RO 0x0	Reserved
2	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 592: USB3_PORTSC Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0420, usb host port1: 0x000F8420
Offset Formula: 0x000F0420+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
1	PED	RW 0x0	PED
0	CCS	RO 0x0	CCS

Table 593: USB3_PORTPMSC Register (n=0–1)
Offset: usb host port0: 0x000F0424, usb host port1: 0x000F8424
Offset Formula: 0x000F0424+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.9.1 xHCI spec Rev1.0			
31:17	Reserved	RO 0x0	Reserved
16	FLA	RW 0x0	FLA
15:8	U2_TIMEOUT	RW 0x0	U2_TIMEOUT Timeout Value for U1 inactivity timer
7:0	U1_TIMEOUT	RW 0x0	U1_TIMEOUT Timeout Value for U1 inactivity timer

Table 594: USB3_PORTLI Register (n=0–1)
Offset: usb host port0: 0x000F0428, usb host port1: 0x000F8428
Offset Formula: 0x000F0428+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.10.1 xHCI spec Rev1.0			
31:16	Reserved	RO 0x0	Reserved
15:0	LEC	RO 0x0	LEC

Table 595: USB3_PORTHLPMC Register (n=0–1)
Offset: usb host port0: 0x000F042C, usb host port1: 0x000F842C
Offset Formula: 0x000F042C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.11.1 xHCI spec Rev1.0			
31:0	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 596: USB2_PORTSC Register (n=0–1)
Offset: usb host port0: 0x000F0430, usb host port1: 0x000F8430
Offset Formula: 0x000F0430+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.8 xHCI spec Rev1.0			
31	Reserved	RO 0x0	Reserved
30	Reserved	RO 0x0	Reserved
29:28	Reserved	RO 0x0	Reserved
27	WOE	RW 0x0	WOE
26	WDE	RW 0x0	WDE
25	WCE	RW 0x0	WCE
24	Reserved	RO 0x0	Reserved
23	CEC	RW 0x0	CEC
22	PLC	RW 0x0	PLC
21	PRC	RW 0x0	PRC
20	Reserved	RO 0x0	Reserved
19	Reserved	RO 0x0	Reserved
18	PEC	RW 0x0	PEC
17	CSC	RW 0x0	CSC
16	LWS	RW 0x0	LWS
15:14	PIC	RW 0x0	PIC
13:10	Port_Speed	RO 0x0	Port_Speed
9	PP	RW 0x1	PP
8:5	PLS	RW 0x5	PLS

Document Classification: Proprietary Information

Table 596: USB2_PORTSC Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F0430, usb host port1: 0x000F8430
Offset Formula: 0x000F0430+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
4	PR	RW 0x0	PR
3	Reserved	RO 0x0	Reserved
2	Reserved	RO 0x0	Reserved
1	PED	RW 0x0	PED
0	CCS	RO 0x0	CCS

Table 597: USB2_PORTPMSC Register (n=0–1)
Offset: usb host port0: 0x000F0434, usb host port1: 0x000F8434
Offset Formula: 0x000F0434+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.9.2 xHCI spec Rev1.0			
31:28	PTC	RW 0x0	PTC
27:17	Reserved	RO 0x0	Reserved
16	HLE	RW 0x0	HLE
15:8	L1_DS	RW 0x0	L1_DS
7:4	BESL	RW 0x0	BESL
3	RWE	RW 0x0	RWE
2:0	L1S	RO 0x0	L1S

Table 598: USB_PORTLI Register (n=0–1)
Offset: usb host port0: 0x000F0438, usb host port1: 0x000F8438
Offset Formula: 0x000F0438+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.10.2 xHCI spec Rev1.0			
31:0	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 599: USB2_PORTHLPMC Register (n=0–1)
Offset: usb host port0: 0x000F043C, usb host port1: 0x000F843C
Offset Formula: 0x000F043C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.4.11.2 xHCI spec Rev1.0			
31:14	Reserved	RO 0x0	Reserved
13:10	BESLD	RW 0x0	BESLD
9:2	L1_TO	RW 0x0	L1_TO
1:0	HIRDM	RW 0x0	HIRDM

A.11.1.3 USB 3.0 Host Controller Runtime

Table 600: MFINDEX Register (n=0–1)
Offset: usb host port0: 0x000F1000, usb host port1: 0x000F9000
Offset Formula: 0x000F1000+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.5.1 xHCI spec Rev1.0			
31:14	Reserved	RO 0x0	Reserved
13:0	MFINDEX	RO 0x0	MFINDEX

Table 601: IMAN Register (n=0–1)
Offset: usb host port0: 0x000F1020, usb host port1: 0x000F9020
Offset Formula: 0x000F1020+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.5.2.1 xHCI spec Rev1.0			
31:2	Reserved	RO 0x0	Reserved
1	IE	RW 0x0	IE
0	IP	RW 0x0	IP

Document Classification: Proprietary Information

Table 602: IMOD Register (n=0–1)
Offset: usb host port0: 0x000F1024, usb host port1: 0x000F9024
Offset Formula: 0x000F1024+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.5.2.2 xHCI spec Rev1.0			
31:16	IMODC	RW 0x0	IMODC
15:0	IMODI	RW 0xFA0	IMODI

Table 603: ERSTSZ Register (n=0–1)
Offset: usb host port0: 0x000F1028, usb host port1: 0x000F9028
Offset Formula: 0x000F1028+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.5.2.3.1 xHCI spec Rev1.0			
31:16	Reserved	RO 0x0	Reserved
15:0	ERSTSZ	RW 0x0	ERSTSZ

Table 604: ERSTBA Register (n=0–1)
Offset: usb host port0: 0x000F1030, usb host port1: 0x000F9030
Offset Formula: 0x000F1030+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.5.2.3.2 xHCI spec Rev1.0			
31:4	ERSTBA_Lo	RW 0x0	ERSTBA_Lo
3:0	Reserved	RO 0x0	Reserved

Table 605: ERDP Register (n=0–1)
Offset: usb host port0: 0x000F1038, usb host port1: 0x000F9038
Offset Formula: 0x000F1038+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.5.2.3.3 xHCI spec Rev1.0			
31:4	ERDQP_Lo	RW 0x0	ERDQP_Lo
3	EHB	RW 0x0	EHB

Copyright © 2015 Marvell

Table 605: ERDP Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F1038, usb host port1: 0x000F9038
Offset Formula: 0x000F1038+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
2:0	DESI	RW 0x0	DESI

A.11.1.4 USB 3.0 Host Controller Doorbell

Table 606: CMD_DB Register (n=0–1)
Offset: usb host port0: 0x000F2000, usb host port1: 0x000FA000
Offset Formula: 0x000F2000+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 5.6 xHCI spec Rev1.0			
31:1	Reserved	RO 0x0	Reserved
0	CMD_DB	WO 0x0	CMD_DB

Table 607: DB1 Register (n=0–1)
Offset: usb host port0: 0x000F2004, usb host port1: 0x000FA004
Offset Formula: 0x000F2004+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 608: DB2 Register (n=0–1)
Offset: usb host port0: 0x000F2008, usb host port1: 0x000FA008
Offset Formula: 0x000F2008+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 609: DB3 Register (n=0–1)
Offset: usb host port0: 0x000F200C, usb host port1: 0x000FA00C
Offset Formula: 0x000F200C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 610: DB4 Register (n=0–1)
Offset: usb host port0: 0x000F2010, usb host port1: 0x000FA010
Offset Formula: 0x000F2010+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 611: DB5 Register (n=0–1)
Offset: usb host port0: 0x000F2014, usb host port1: 0x000FA014
Offset Formula: 0x000F2014+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 612: DB6 Register (n=0–1)
Offset: usb host port0: 0x000F2018, usb host port1: 0x000FA018
Offset Formula: 0x000F2018+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 613: DB7 Register (n=0–1)
Offset: usb host port0: 0x000F201C, usb host port1: 0x000FA01C
Offset Formula: 0x000F201C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 614: DB8 Register (n=0–1)
Offset: usb host port0: 0x000F2020, usb host port1: 0x000FA020
Offset Formula: 0x000F2020+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 615: DB9 Register (n=0–1)
Offset: usb host port0: 0x000F2024, usb host port1: 0x000FA024
Offset Formula: 0x000F2024+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 616: DB10 Register (n=0–1)
Offset: usb host port0: 0x000F2028, usb host port1: 0x000FA028
Offset Formula: 0x000F2028+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 617: DB11 Register (n=0–1)
Offset: usb host port0: 0x000F202C, usb host port1: 0x000FA02C
Offset Formula: 0x000F202C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 618: DB12 Register (n=0–1)
Offset: usb host port0: 0x000F2030, usb host port1: 0x000FA030
Offset Formula: 0x000F2030+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 619: DB13 Register (n=0–1)
Offset: usb host port0: 0x000F2034, usb host port1: 0x000FA034
Offset Formula: 0x000F2034+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 620: DB14 Register (n=0–1)
Offset: usb host port0: 0x000F2038, usb host port1: 0x000FA038
Offset Formula: 0x000F2038+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 621: DB15 Register (n=0–1)
Offset: usb host port0: 0x000F203C, usb host port1: 0x000FA03C
Offset Formula: 0x000F203C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 622: DB16 Register (n=0–1)
Offset: usb host port0: 0x000F2040, usb host port1: 0x000FA040
Offset Formula: 0x000F2040+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 623: DB17 Register (n=0–1)
Offset: usb host port0: 0x000F2044, usb host port1: 0x000FA044
Offset Formula: 0x000F2044+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 624: DB18 Register (n=0–1)
Offset: usb host port0: 0x000F2048, usb host port1: 0x000FA048
Offset Formula: 0x000F2048+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 625: DB19 Register (n=0–1)
Offset: usb host port0: 0x000F204C, usb host port1: 0x000FA04C
Offset Formula: 0x000F204C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 626: DB20 Register (n=0–1)
Offset: usb host port0: 0x000F2050, usb host port1: 0x000FA050
Offset Formula: 0x000F2050+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 627: DB21 Register (n=0–1)
Offset: usb host port0: 0x000F2054, usb host port1: 0x000FA054
Offset Formula: 0x000F2054+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 628: DB22 Register (n=0–1)
Offset: usb host port0: 0x000F2058, usb host port1: 0x000FA058
Offset Formula: 0x000F2058+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 629: DB23 Register (n=0–1)
Offset: usb host port0: 0x000F205C, usb host port1: 0x000FA05C
Offset Formula: 0x000F205C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 630: DB24 Register (n=0–1)
Offset: usb host port0: 0x000F2060, usb host port1: 0x000FA060
Offset Formula: 0x000F2060+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 631: DB25 Register (n=0–1)
Offset: usb host port0: 0x000F2064, usb host port1: 0x000FA064
Offset Formula: 0x000F2064+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 632: DB26 Register (n=0–1)
Offset: usb host port0: 0x000F2068, usb host port1: 0x000FA068
Offset Formula: 0x000F2068+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 633: DB27 Register (n=0–1)
Offset: usb host port0: 0x000F206C, usb host port1: 0x000FA06C
Offset Formula: 0x000F206C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 634: DB28 Register (n=0–1)
Offset: usb host port0: 0x000F2070, usb host port1: 0x000FA070
Offset Formula: 0x000F2070+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 635: DB29 Register (n=0–1)
Offset: usb host port0: 0x000F2074, usb host port1: 0x000FA074
Offset Formula: 0x000F2074+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Table 636: DB30 Register (n=0–1)
Offset: usb host port0: 0x000F2078, usb host port1: 0x000FA078
Offset Formula: 0x000F2078+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

Copyright © 2015 Marvell

Table 637: DB31 Register (n=0–1)
Offset: usb host port0: 0x000F207C, usb host port1: 0x000FA07C
Offset Formula: 0x000F207C+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved
4:0	EP_NUM	RW 0x0	EP_NUM

A.11.1.5 USB 3.0 Host Controller Extended Capability

Table 638: xECP_SUP_CAP_USB2_R1 Register (n=0–1)
Offset: usb host port0: 0x000F2800, usb host port1: 0x000FA800
Offset Formula: 0x000F2800+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
Reference: Section 7.2 xHCI spec Rev1.0			
31:24	MAJ_REV	RO 0x2	MIJ_REV
23:16	MIN_REV	RO 0x0	MIN_REV
15:8	NXT_ECP	RO 0x4	NXT_ECP
7:0	CAP_ID	RO 0x2	CAP_ID

Table 639: xECP_SUP_CAP_USB2_R2 Register (n=0–1)
Offset: usb host port0: 0x000F2804, usb host port1: 0x000FA804
Offset Formula: 0x000F2804+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:0	NAME_STRING	RO 0x20425355	NAME_STRING

Table 640: xECP_SUP_CAP_USB2_R3 Register (n=0–1)
Offset: usb host port0: 0x000F2808, usb host port1: 0x000FA808
Offset Formula: 0x000F2808+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:28	PSIC	RO 0x0	PSIC
27:20	Reserved	RO 0x0	Reserved

Document Classification: Proprietary Information

Table 640: xECP_SUP_CAP_USB2_R3 Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F2808, usb host port1: 0x000FA808
Offset Formula: 0x000F2808+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
19	HLC	RO 0x0	HLC
18	IHI	RO 0x0	IHI
17	HSO	RO 0x0	HSO
16	Reserved	RO 0x1	Reserved
15:8	CP_COUNT	RO 0x1	CP_COUNT
7:0	CP_OFFSET	RO 0x2	CP_OFFSET

Table 641: xECP_SUP_CAP_USB3_R1 Register (n=0–1)
Offset: usb host port0: 0x000F2810, usb host port1: 0x000FA810
Offset Formula: 0x000F2810+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:24	MAJ_REV	RO 0x3	MAJ_REV
23:16	MIN_REV	RO 0x0	MIN_REV
15:8	NXT_ECP	RO 0x0	NXT_ECP
7:0	CAP_ID	RO 0x2	CAP_ID

Table 642: xECP_SUP_CAP_USB3_R2 Register (n=0–1)
Offset: usb host port0: 0x000F2814, usb host port1: 0x000FA814
Offset Formula: 0x000F2814+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:0	NAME_STRING	RO 0x20425355	NAME_STRING

Table 643: xECP_SUP_CAP_USB3_R3 Register (n=0–1)
Offset: usb host port0: 0x000F2818, usb host port1: 0x000FA818
Offset Formula: 0x000F2818+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:28	PSIC	RO 0x0	PSIC
27:20	Reserved	RO 0x0	Reserved
19	HLC	RO 0x0	HLC
18	IHI	RO 0x0	IHI
17	HSO	RO 0x0	HSO
16	Reserved	RO 0x0	Reserved
15:8	CP_COUNT	RO 0x1	CP_COUNT
7:0	CP_OFFSET	RO 0x1	CP_OFFSET

Copyright © 2015 Marvell

A.12 USB 3.0 Host Controller Bridge Registers

The following table provides a summarized list of all registers that belong to the USB 3.0 Host Controller Bridge, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 644: Summary Map Table for the USB 3.0 Host Controller Bridge Registers

Register Name	Offset	Table and Page
Bridge Window Control Register (n=0–1, N=0–7)	usb host port0Windows0: 0x000F4000, usb host port0Windows1: 0x000F4008...usb host port1Windows7: 0x000FC038	Table 645, p. 745
Bridge Window Base Register (n=0–1, N=0–7)	usb host port0Windows0: 0x000F4004, usb host port0Windows1: 0x000F400C...usb host port1Windows7: 0x000FC03C	Table 646, p. 746
Mbus Control Register (n=0–1)	usb host port0: 0x000F4040, usb host port1: 0x000FC040	Table 647, p. 746
Sync Barrier WindowN Control Register (n=0–1, N=0–1)	usb host port0SyncbWin0: 0x000F4044, usb host port0SyncbWin1: 0x000F404C, usb host port1SyncbWin0: 0x000FC044, usb host port1SyncbWin1: 0x000FC04C	Table 648, p. 747
Sync Barrier WindowN Base Register (n=0–1, N=0–1)	usb host port0SyncbWin0: 0x000F4048, usb host port0SyncbWin1: 0x000F4050, usb host port1SyncbWin0: 0x000FC048, usb host port1SyncbWin1: 0x000FC050	Table 649, p. 747

Table 645: Bridge Window Control Register (n=0–1, N=0–7)
Offset: usb host port0Windows0: 0x000F4000, usb host port0Windows1: 0x000F4008...usb host port1Windows7: 0x000FC038
Offset Formula: 0x000F4000+0x8*N + 0x8000*n : where n (0-1) represents usb host port, where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 64 KB granularity (for example, a value of 0x00FF specifies 256*64KB = 16 MB). NOTES: A value of 0x0 specifies 64-KB size. Changing the value of this field must be done when <Win En>=0.

Table 645: Bridge Window Control Register (n=0–1, N=0–7) (Continued)
Offset: usb host port0Windows0: 0x000F4000, usb host port0Windows1: 0x000F4008...usb host port1Windows7: 0x000FC038
Offset Formula: 0x000F4000+0x8*N + 0x8000*n : where n (0-1) represents usb host port, where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this window. See the "Target Interface Configurations" section. Changing the value of this field must be done when <Win En>=0.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. See the "Target Interface Configurations" section. Changing the value of this field must be done when <Win En>=0.
3:1	Reserved	RSVD 0x0	Reserved
0	Win En	RW 0x0	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Table 646: Bridge Window Base Register (n=0–1, N=0–7)
Offset: usb host port0Windows0: 0x000F4004, usb host port0Windows1: 0x000F400C...usb host port1Windows7: 0x000FC03C
Offset Formula: 0x000F4004+0x8*N + 0x8000*n : where n (0-1) represents usb host port, where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register!			
31:16	Base	RW 0x0	Base Address Used with the <Size> field in the Window0 Control Register to set the address window size and location. Corresponds to transaction address[31:16]. Changing the value of this field must be done when <Win0 En>=0
15:0	Reserved	RSVD 0x0	Reserved

Table 647: Mbus Control Register (n=0–1)
Offset: usb host port0: 0x000F4040, usb host port1: 0x000FC040
Offset Formula: 0x000F4040+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8:1	Mbus Read Regret Timer	RW 0x40	Mbus Read Regret Timer. Mbus read request may regret when timer expires and Mbus Regret is Enabled.

Copyright © 2015 Marvell

Table 647: Mbus Control Register (n=0–1) (Continued)
Offset: usb host port0: 0x000F4040, usb host port1: 0x000FC040
Offset Formula: 0x000F4040+0x8000*n: where n (0-1) represents usb host port

Bit	Field	Type/InitVal	Description
0	Mbus Read Regret Enable	RW 0x0	Allows the Mbus read request to regret when the Regret Timer expires.

Table 648: Sync Barrier WindowN Control Register (n=0–1, N=0–1)
Offset: usb host port0SyncbWin0: 0x000F4044, usb host port0SyncbWin1: 0x000F404C, usb host port1SyncbWin0: 0x000FC044, usb host port1SyncbWin1: 0x000FC04C
Offset Formula: 0x000F4044+0x8*N + 0x8000*n : where n (0-1) represents usb host port, where N (0-1) represents SyncbWin

Bit	Field	Type/InitVal	Description
31:2	Sync Barrier WindowN Size	RW 0x0	Sync Barrier Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 4 B granularity (for example, a value of 0x00FF specifies 256*4B = 1KB). NOTES: A value of 0x0 specifies 64-KB size. Changing the value of this field must be done when <Win En>=0
1	Reserved	RSVD 0x0	Reserved
0	Sync Barrier WindowN Enable	RW 0x0	Window Sync Barrier Enable When enabled, data associated with read requests that hit the window will be returned to the requesting CPU after all writes issued by the target unit reach the memory. 0 = Disable: Non blocking 1 = Enable: Blocking

Table 649: Sync Barrier WindowN Base Register (n=0–1, N=0–1)
Offset: usb host port0SyncbWin0: 0x000F4048, usb host port0SyncbWin1: 0x000F4050, usb host port1SyncbWin0: 0x000FC048, usb host port1SyncbWin1: 0x000FC050
Offset Formula: 0x000F4048+0x8*N + 0x8000*n : where n (0-1) represents usb host port, where N (0-1) represents SyncbWin

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register!			
31:2	Sync Barrier WindowN Base	RW 0x0	Base Address Used with the <Size> field in the Sync Barrier Window Control Register to set the address window size and location. Corresponds to transaction address[31:2]. Changing the value of this field must be done when <Sync Barrier Enable>=0
1:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.13 USB 3.0 Device Registers

The following table provides a summarized list of all registers that belong to the USB 3.0 Device, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 650: Summary Map Table for the USB 3.0 Device Registers

Register Name	Offset	Table and Page
USB 3.0 Device Controller		
MAC		
TSEQ_COUNTER_U1U2_TIMEOUT_REG Register	0x00050190	Table 651, p. 763
TIMER_TIMEOUT_REG_1 Register	0x00050194	Table 652, p. 764
TIMER_TIMEOUT_REG_2 Register	0x00050198	Table 653, p. 764
LFPS_TX_CONFIG_REG Register	0x0005019C	Table 654, p. 764
LFPS_RX_CONFIG_REG Register	0x000501A0	Table 655, p. 765
LFPS_WR_TRESET_VAL_REG Register	0x000501A4	Table 656, p. 765
COUNTER_DELAY_TX_REG Register	0x000501A8	Table 657, p. 766
PHY_CONFIG_REG Register	0x000501AC	Table 658, p. 766
LOOP_BACK_REG Register	0x000501B0	Table 659, p. 767
BERT_INTERVAL_REG Register	0x000501B4	Table 660, p. 767
SHALLOW_LOOP_BACK_REG Register	0x000501B8	Table 661, p. 768
Status		
POWER_MANAGEMENT_DEVICE_REG Register	0x000500C8	Table 662, p. 768
POWER_MANAGEMENT_SOC_REG Register	0x000500CC	Table 663, p. 769
LOW_POWER_STATUS_REG Register	0x000500D0	Table 664, p. 770
SOFTWARE_RESET_REG Register	0x000500D4	Table 665, p. 770
TOP_INTERRUPT_STATUS_REG Register	0x000500D8	Table 666, p. 771
TOP_INTERRUPT_ENABLE_REG Register	0x000500DC	Table 667, p. 772
PME_INTERRUPT_STATUS_EN_REG Register	0x000500E0	Table 668, p. 772
AHB_TIMEOUT_REG Register	0x000500E4	Table 669, p. 773
PA		
DSP_DISCONNECT Register	0x00058400	Table 670, p. 773
CONFIGURE_FOR_HS Register	0x00058401	Table 671, p. 774
BURST_CLOSURE Register	0x00058402	Table 672, p. 774
DISABLE_SCRAMBLING Register	0x00058403	Table 673, p. 775
DISABLE_STALL_IN_U0 Register	0x00058404	Table 674, p. 776
DISABLE_LUP_LDN Register	0x00058405	Table 675, p. 776
TEST_MODE Register	0x000584FF	Table 676, p. 777
PAIR_CAPABILITY Register	0x00058500	Table 677, p. 778
LOOPBACK_EN Register	0x00058E00	Table 678, p. 778
RX_BURST_COUNT_0 Register	0x00058E01	Table 679, p. 779
RX_BURST_COUNT_1 Register	0x00058E02	Table 680, p. 780
RX_BURST_COUNT_2 Register	0x00058E03	Table 681, p. 780

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
RX_BURST_COUNT_3 Register	0x00058E04	Table 682, p. 781
RX_ERR_COUNT_0 Register	0x00058E05	Table 683, p. 781
RX_ERR_COUNT_1 Register	0x00058E06	Table 684, p. 782
RX_ERR_COUNT_2 Register	0x00058E07	Table 685, p. 782
RX_ERR_COUNT_3 Register	0x00058E08	Table 686, p. 783
RX_COUNT_RESET Register	0x00058E09	Table 687, p. 783
TX_COMP_MODE_EN Register	0x00058E0A	Table 688, p. 784
ANALOG_LOOPBACK_EN Register	0x00058E0B	Table 689, p. 784
PM_CLOCK_GATE_EN Register	0x00058F00	Table 690, p. 785
Timer_Hibern8_Enter_TX Register	0x00058F02	Table 691, p. 785
Timer_Hibern8_Residency Register	0x00058F04	Table 692, p. 786
Num_of_Cycles_for_1us Register	0x00058F06	Table 693, p. 786
MPHY_RMMI_Register_Base_Addr	0x00058F08	Table 694, p. 786
Num_of_Cycles_for_10us Register	0x00058F0A	Table 695, p. 787
SSIC_Dev_Timer_Tretrain Register	0x00058F0E	Table 696, p. 787
SSIC_Dev_Pollingactive_Residency Register	0x00058F10	Table 697, p. 787
SSIC_Control_Status_Register	0x00058F20	Table 698, p. 788
SSIC_Status_Interrupt_Enable_Register	0x00058F22	Table 699, p. 789
Timer_Tx_Tactive_Time Register	0x00058F28	Table 700, p. 790
Timer_Rx_Tactive_Time Register	0x00058F2A	Table 701, p. 790
Reference		
DEV_INFO_REG Register	0x00050000	Table 702, p. 791
DEV_TAG_REG Register	0x00050004	Table 703, p. 791
LOW_POWER_LFPS_SIGNAL_VAL_REG_1 Register	0x00050008	Table 704, p. 792
LOW_POWER_LFPS_SIGNAL_VAL_REG_2 Register	0x0005000C	Table 705, p. 792
LOW_POWER_LFPS_SIGNAL_VAL_REG_3 Register	0x00050010	Table 706, p. 793
LOW_POWER_LFPS_SIGNAL_VAL_REG_4 Register	0x00050014	Table 707, p. 793
LOW_POWER_LFPS_SIGNAL_VAL_REG_5 Register	0x00050018	Table 708, p. 793
LOW_POWER_LFPS_SIGNAL_VAL_REG_6 Register	0x0005001C	Table 709, p. 794
COUNTER_PULSE_VALUE_REG Register	0x00050020	Table 710, p. 794
REF_INT_STATUS_REG Register	0x00050024	Table 711, p. 794
REF_INT_ENABLE_REG Register	0x00050028	Table 712, p. 795
USB3_GLOBAL_CONTROL_STATUS_REG Register	0x0005002C	Table 713, p. 796
SYSTEM_DEBUG_REG Register	0x00050030	Table 714, p. 797
EP0_ACK_TIMEOUT_VALUE_REG Register	0x00050034	Table 715, p. 799
ACK_TIMEOUT_VALUE_REG Register	0x00050038	Table 716, p. 800
RX_IDLE_DELAY_COUNTER_REG Register	0x0005003C	Table 717, p. 800
COUNTER_PULSE_VALUE_REG_1 Register	0x00050040	Table 718, p. 800
TEST_BUS_MUX_REG Register	0x00050044	Table 719, p. 801
CLOCK_GATING_ENABLE_REG Register	0x00050048	Table 720, p. 801
POWER_GATING_ENABLE_REG Register	0x0005004C	Table 721, p. 802

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
Core and Endpoint		
DEV_USB_ADDRESS_REG Register	0x00050320	Table 722, p. 803
FUNCTION_WAKEUP_REG Register	0x00050324	Table 723, p. 803
ENDPOINT_0_CONFIG_REG Register	0x00050328	Table 724, p. 803
OUT_ENDPOINT_CONFIG_REG_1 Register	0x0005032C	Table 725, p. 804
OUT_ENDPOINT_CONFIG_REG_2 Register	0x00050330	Table 726, p. 806
OUT_ENDPOINT_CONFIG_REG_3 Register	0x00050334	Table 727, p. 807
OUT_ENDPOINT_CONFIG_REG_4 Register	0x00050338	Table 728, p. 808
OUT_ENDPOINT_CONFIG_REG_5 Register	0x0005033C	Table 729, p. 810
OUT_ENDPOINT_CONFIG_REG_6 Register	0x00050340	Table 730, p. 811
OUT_ENDPOINT_CONFIG_REG_7 Register	0x00050344	Table 731, p. 813
OUT_ENDPOINT_CONFIG_REG_8 Register	0x00050348	Table 732, p. 814
OUT_ENDPOINT_CONFIG_REG_9 Register	0x0005034C	Table 733, p. 816
OUT_ENDPOINT_CONFIG_REG_10 Register	0x00050350	Table 734, p. 817
OUT_ENDPOINT_CONFIG_REG_11 Register	0x00050354	Table 735, p. 819
OUT_ENDPOINT_CONFIG_REG_12 Register	0x00050358	Table 736, p. 820
OUT_ENDPOINT_CONFIG_REG_13 Register	0x0005035C	Table 737, p. 822
OUT_ENDPOINT_CONFIG_REG_14 Register	0x00050360	Table 738, p. 823
OUT_ENDPOINT_CONFIG_REG_15 Register	0x00050364	Table 739, p. 825
IN_ENDPOINT_CONFIG_REG_1 Register	0x00050368	Table 740, p. 826
IN_ENDPOINT_CONFIG_REG_2 Register	0x0005036C	Table 741, p. 828
IN_ENDPOINT_CONFIG_REG_3 Register	0x00050370	Table 742, p. 829
IN_ENDPOINT_CONFIG_REG_4 Register	0x00050374	Table 743, p. 831
IN_ENDPOINT_CONFIG_REG_5 Register	0x00050378	Table 744, p. 832
IN_ENDPOINT_CONFIG_REG_6 Register	0x0005037C	Table 745, p. 834
IN_ENDPOINT_CONFIG_REG_7 Register	0x00050380	Table 746, p. 835
IN_ENDPOINT_CONFIG_REG_8 Register	0x00050384	Table 747, p. 837
IN_ENDPOINT_CONFIG_REG_9 Register	0x00050388	Table 748, p. 838
IN_ENDPOINT_CONFIG_REG_10 Register	0x0005038C	Table 749, p. 840
IN_ENDPOINT_CONFIG_REG_11 Register	0x00050390	Table 750, p. 841
IN_ENDPOINT_CONFIG_REG_12 Register	0x00050394	Table 751, p. 843
IN_ENDPOINT_CONFIG_REG_13 Register	0x00050398	Table 752, p. 844
IN_ENDPOINT_CONFIG_REG_14 Register	0x0005039C	Table 753, p. 846
IN_ENDPOINT_CONFIG_REG_15 Register	0x000503A0	Table 754, p. 847
PHY_CFG_REG Register	0x000503A4	Table 755, p. 849
BUS_INTERVAL_LTM_REG Register	0x000503A8	Table 756, p. 851
ISO_TIME_STAMP_COUNTER_REG Register	0x000503AC	Table 757, p. 851
CREDIT_HP_AND_U2_TIMER_REG Register	0x000503B0	Table 758, p. 851
PM_LC_TIMER_TIME_OUT_REG Register	0x000503B4	Table 759, p. 852
SS_CORE_INTERRUPT_REG Register	0x000503B8	Table 760, p. 852
SS_CORE_INTERRUPT_EN_REG Register	0x000503BC	Table 761, p. 854

Copyright © 2015 Marvell
Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_BINTERVAL_REG_1_2_3 Register	0x000503C0	Table 762, p. 855
EP_IN_BINTERVAL_REG_4_5_6_7 Register	0x000503C4	Table 763, p. 855
EP_IN_BINTERVAL_REG_8_9_10_11 Register	0x000503C8	Table 764, p. 855
EP_IN_BINTERVAL_REG_12_13_14_15 Register	0x000503CC	Table 765, p. 856
EP_OUT_BINTERVAL_REG_1_2_3 Register	0x000503D0	Table 766, p. 856
EP_OUT_BINTERVAL_REG_4_5_6_7 Register	0x000503D4	Table 767, p. 857
EP_OUT_BINTERVAL_REG_8_9_10_11 Register	0x000503D8	Table 768, p. 857
EP_OUT_BINTERVAL_REG_12_13_14_15 Register	0x000503DC	Table 769, p. 858
EP_HOST_ERROR_STATUS_REG Register	0x000503E0	Table 770, p. 858
UX_EXIT_DETECTION_DELAY_REG Register	0x000503E8	Table 771, p. 859
TX_TSI_NUM_REG Register	0x000503EC	Table 772, p. 860
START_STATE_DELAY_AND_MIN_TS2_HOT_RESET_REG Register	0x000503F0	Table 773, p. 860
LOWPOWER_MANAGEMENT_REG Register	0x000503F4	Table 774, p. 861
USB3_SETUP_DP_LOW Register	0x000503F8	Table 775, p. 861
USB3_SETUP_DP_HIGH_REG Register	0x000503FC	Table 776, p. 861
USB3_SETUP_CONTROL_REG Register	0x00050400	Table 777, p. 861
DELAY_U1_CNT_INIT_VALUE_REG Register	0x00050404	Table 778, p. 862
DELAY_U2_CNT_INIT_VALUE_REG Register	0x00050408	Table 779, p. 862
NO_BURST_MODE_REG Register	0x0005040C	Table 780, p. 862
System		
DMA_GLOBAL_CONFIG_REG Register	0x000507D0	Table 781, p. 862
BULK_STREAMING_ENABLE_REG Register	0x000507D4	Table 782, p. 863
EP_OUT_RCV_STREAM_ID_REG_1 Register	0x000507D8	Table 783, p. 865
EP_OUT_RCV_STREAM_ID_REG_2 Register	0x000507DC	Table 784, p. 865
EP_OUT_RCV_STREAM_ID_REG_3 Register	0x000507E0	Table 785, p. 866
EP_OUT_RCV_STREAM_ID_REG_4 Register	0x000507E4	Table 786, p. 866
EP_OUT_RCV_STREAM_ID_REG_5 Register	0x000507E8	Table 787, p. 866
EP_OUT_RCV_STREAM_ID_REG_6 Register	0x000507EC	Table 788, p. 866
EP_OUT_RCV_STREAM_ID_REG_7 Register	0x000507F0	Table 789, p. 867
EP_OUT_RCV_STREAM_ID_REG_8 Register	0x000507F4	Table 790, p. 867
EP_OUT_RCV_STREAM_ID_REG_9 Register	0x000507F8	Table 791, p. 867
EP_OUT_RCV_STREAM_ID_REG_10 Register	0x000507FC	Table 792, p. 867
EP_OUT_RCV_STREAM_ID_REG_11 Register	0x00050800	Table 793, p. 868
EP_OUT_RCV_STREAM_ID_REG_12 Register	0x00050804	Table 794, p. 868
EP_OUT_RCV_STREAM_ID_REG_13 Register	0x00050808	Table 795, p. 868
EP_OUT_RCV_STREAM_ID_REG_14 Register	0x0005080C	Table 796, p. 868
EP_OUT_RCV_STREAM_ID_REG_15 Register	0x00050810	Table 797, p. 869
EP_IN_RCV_STREAM_ID_REG_1 Register	0x00050814	Table 798, p. 869
EP_IN_RCV_STREAM_ID_REG_2 Register	0x00050818	Table 799, p. 869
EP_IN_RCV_STREAM_ID_REG_3 Register	0x0005081C	Table 800, p. 869

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_RCV_STREAM_ID_REG_4 Register	0x00050820	Table 801, p. 870
EP_IN_RCV_STREAM_ID_REG_5 Register	0x00050824	Table 802, p. 870
EP_IN_RCV_STREAM_ID_REG_6 Register	0x00050828	Table 803, p. 870
EP_IN_RCV_STREAM_ID_REG_7 Register	0x0005082C	Table 804, p. 870
EP_IN_RCV_STREAM_ID_REG_8 Register	0x00050830	Table 805, p. 871
EP_IN_RCV_STREAM_ID_REG_9 Register	0x00050834	Table 806, p. 871
EP_IN_RCV_STREAM_ID_REG_10 Register	0x00050838	Table 807, p. 871
EP_IN_RCV_STREAM_ID_REG_11 Register	0x0005083C	Table 808, p. 871
EP_IN_RCV_STREAM_ID_REG_12 Register	0x00050840	Table 809, p. 872
EP_IN_RCV_STREAM_ID_REG_13 Register	0x00050844	Table 810, p. 872
EP_IN_RCV_STREAM_ID_REG_14 Register	0x00050848	Table 811, p. 872
EP_IN_RCV_STREAM_ID_REG_15 Register	0x0005084C	Table 812, p. 872
DMA_COMPLETE_SUCCESS_STATUS Register	0x00050850	Table 813, p. 873
DMA_COMPLETE_ERROR_STATUS Register	0x00050854	Table 814, p. 878
DMA_BD_FETCH_ERROR_STATUS_REG Register	0x00050858	Table 815, p. 883
DMA_BD_FETCH_ERROR_EN_REG Register	0x0005085C	Table 816, p. 885
DMA_DATA_ERROR_STATUS_REG Register	0x00050860	Table 817, p. 886
DMA_DATA_ERROR_EN_REG Register	0x00050864	Table 818, p. 888
DMA_ERROR_HANDLING_MODE_REG Register	0x00050868	Table 819, p. 890
EP_OUT_RX_DMA_CONFIG_REG_0 Register	0x0005086C	Table 820, p. 890
EP_OUT_RX_DMA_CONFIG_REG_1 Register	0x00050870	Table 821, p. 891
EP_OUT_RX_DMA_CONFIG_REG_2 Register	0x00050874	Table 822, p. 891
EP_OUT_RX_DMA_CONFIG_REG_3 Register	0x00050878	Table 823, p. 892
EP_OUT_RX_DMA_CONFIG_REG_4 Register	0x0005087C	Table 824, p. 892
EP_OUT_RX_DMA_CONFIG_REG_5 Register	0x00050880	Table 825, p. 893
EP_OUT_RX_DMA_CONFIG_REG_6 Register	0x00050884	Table 826, p. 893
EP_OUT_RX_DMA_CONFIG_REG_7 Register	0x00050888	Table 827, p. 894
EP_OUT_RX_DMA_CONFIG_REG_8 Register	0x0005088C	Table 828, p. 894
EP_OUT_RX_DMA_CONFIG_REG_9 Register	0x00050890	Table 829, p. 895
EP_OUT_RX_DMA_CONFIG_REG_10 Register	0x00050894	Table 830, p. 895
EP_OUT_RX_DMA_CONFIG_REG_11 Register	0x00050898	Table 831, p. 896
EP_OUT_RX_DMA_CONFIG_REG_12 Register	0x0005089C	Table 832, p. 896
EP_OUT_RX_DMA_CONFIG_REG_13 Register	0x000508A0	Table 833, p. 897
EP_OUT_RX_DMA_CONFIG_REG_14 Register	0x000508A4	Table 834, p. 897
EP_OUT_RX_DMA_CONFIG_REG_15 Register	0x000508A8	Table 835, p. 898
EP_IN_TX_DMA_CONFIG_REG_0 Register	0x000508AC	Table 836, p. 898
EP_IN_TX_DMA_CONFIG_REG_1 Register	0x000508B0	Table 837, p. 899
EP_IN_TX_DMA_CONFIG_REG_2 Register	0x000508B4	Table 838, p. 899
EP_IN_TX_DMA_CONFIG_REG_3 Register	0x000508B8	Table 839, p. 900
EP_IN_TX_DMA_CONFIG_REG_4 Register	0x000508BC	Table 840, p. 900
EP_IN_TX_DMA_CONFIG_REG_5 Register	0x000508C0	Table 841, p. 901

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_TX_DMA_CONFIG_REG_6 Register	0x000508C4	Table 842, p. 901
EP_IN_TX_DMA_CONFIG_REG_7 Register	0x000508C8	Table 843, p. 902
EP_IN_TX_DMA_CONFIG_REG_8 Register	0x000508CC	Table 844, p. 902
EP_IN_TX_DMA_CONFIG_REG_9 Register	0x000508D0	Table 845, p. 903
EP_IN_TX_DMA_CONFIG_REG_10 Register	0x000508D4	Table 846, p. 903
EP_IN_TX_DMA_CONFIG_REG_11 Register	0x000508D8	Table 847, p. 904
EP_IN_TX_DMA_CONFIG_REG_12 Register	0x000508DC	Table 848, p. 904
EP_IN_TX_DMA_CONFIG_REG_13 Register	0x000508E0	Table 849, p. 905
EP_IN_TX_DMA_CONFIG_REG_14 Register	0x000508E4	Table 850, p. 905
EP_IN_TX_DMA_CONFIG_REG_15 Register	0x000508E8	Table 851, p. 906
EP_OUT_RX_DMA_START_ADDRESS_REG_0 Register	0x000508EC	Table 852, p. 906
EP_OUT_RX_DMA_START_ADDRESS_REG_1 Register	0x000508F0	Table 853, p. 907
EP_OUT_RX_DMA_START_ADDRESS_REG_2 Register	0x000508F4	Table 854, p. 907
EP_OUT_RX_DMA_START_ADDRESS_REG_3 Register	0x000508F8	Table 855, p. 907
EP_OUT_RX_DMA_START_ADDRESS_REG_4 Register	0x000508FC	Table 856, p. 907
EP_OUT_RX_DMA_START_ADDRESS_REG_5 Register	0x00050900	Table 857, p. 907
EP_OUT_RX_DMA_START_ADDRESS_REG_6 Register	0x00050904	Table 858, p. 908
EP_OUT_RX_DMA_START_ADDRESS_REG_7 Register	0x00050908	Table 859, p. 908
EP_OUT_RX_DMA_START_ADDRESS_REG_8 Register	0x0005090C	Table 860, p. 908
EP_OUT_RX_DMA_START_ADDRESS_REG_9 Register	0x00050910	Table 861, p. 908
EP_OUT_RX_DMA_START_ADDRESS_REG_10 Register	0x00050914	Table 862, p. 908
EP_OUT_RX_DMA_START_ADDRESS_REG_11 Register	0x00050918	Table 863, p. 909
EP_OUT_RX_DMA_START_ADDRESS_REG_12 Register	0x0005091C	Table 864, p. 909
EP_OUT_RX_DMA_START_ADDRESS_REG_13 Register	0x00050920	Table 865, p. 909
EP_OUT_RX_DMA_START_ADDRESS_REG_14 Register	0x00050924	Table 866, p. 909
EP_OUT_RX_DMA_START_ADDRESS_REG_15 Register	0x00050928	Table 867, p. 909
EP_IN_TX_DMA_START_ADDRESS_REG_0 Register	0x0005092C	Table 868, p. 910
EP_IN_TX_DMA_START_ADDRESS_REG_1 Register	0x00050930	Table 869, p. 910
EP_IN_TX_DMA_START_ADDRESS_REG_2 Register	0x00050934	Table 870, p. 910
EP_IN_TX_DMA_START_ADDRESS_REG_3 Register	0x00050938	Table 871, p. 910

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_TX_DMA_START_ADDRESS_REG_4 Register	0x0005093C	Table 872, p. 910
EP_IN_TX_DMA_START_ADDRESS_REG_5 Register	0x00050940	Table 873, p. 911
EP_IN_TX_DMA_START_ADDRESS_REG_6 Register	0x00050944	Table 874, p. 911
EP_IN_TX_DMA_START_ADDRESS_REG_7 Register	0x00050948	Table 875, p. 911
EP_IN_TX_DMA_START_ADDRESS_REG_8 Register	0x0005094C	Table 876, p. 911
EP_IN_TX_DMA_START_ADDRESS_REG_9 Register	0x00050950	Table 877, p. 911
EP_IN_TX_DMA_START_ADDRESS_REG_10 Register	0x00050954	Table 878, p. 912
EP_IN_TX_DMA_START_ADDRESS_REG_11 Register	0x00050958	Table 879, p. 912
EP_IN_TX_DMA_START_ADDRESS_REG_12 Register	0x0005095C	Table 880, p. 912
EP_IN_TX_DMA_START_ADDRESS_REG_13 Register	0x00050960	Table 881, p. 912
EP_IN_TX_DMA_START_ADDRESS_REG_14 Register	0x00050964	Table 882, p. 912
EP_IN_TX_DMA_START_ADDRESS_REG_15 Register	0x00050968	Table 883, p. 913
EP_OUT_DMA_ERROR_ADDRESS_REG_0 Register	0x0005096C	Table 884, p. 913
EP_OUT_DMA_ERROR_ADDRESS_REG_1 Register	0x00050970	Table 885, p. 913
EP_OUT_DMA_ERROR_ADDRESS_REG_2 Register	0x00050974	Table 886, p. 913
EP_OUT_DMA_ERROR_ADDRESS_REG_3 Register	0x00050978	Table 887, p. 914
EP_OUT_DMA_ERROR_ADDRESS_REG_4 Register	0x0005097C	Table 888, p. 914
EP_OUT_DMA_ERROR_ADDRESS_REG_5 Register	0x00050980	Table 889, p. 914
EP_OUT_DMA_ERROR_ADDRESS_REG_6 Register	0x00050984	Table 890, p. 915
EP_OUT_DMA_ERROR_ADDRESS_REG_7 Register	0x00050988	Table 891, p. 915
EP_OUT_DMA_ERROR_ADDRESS_REG_8 Register	0x0005098C	Table 892, p. 915
EP_OUT_DMA_ERROR_ADDRESS_REG_9 Register	0x00050990	Table 893, p. 916
EP_OUT_DMA_ERROR_ADDRESS_REG_10 Register	0x00050994	Table 894, p. 916
EP_OUT_DMA_ERROR_ADDRESS_REG_11 Register	0x00050998	Table 895, p. 916
EP_OUT_DMA_ERROR_ADDRESS_REG_12 Register	0x0005099C	Table 896, p. 917
EP_OUT_DMA_ERROR_ADDRESS_REG_13 Register	0x000509A0	Table 897, p. 917
EP_OUT_DMA_ERROR_ADDRESS_REG_14 Register	0x000509A4	Table 898, p. 917
EP_OUT_DMA_ERROR_ADDRESS_REG_15 Register	0x000509A8	Table 899, p. 918
EP_IN_DMA_ERROR_ADDRESS_REG_0 Register	0x000509AC	Table 900, p. 918
EP_IN_DMA_ERROR_ADDRESS_REG_1 Register	0x000509B0	Table 901, p. 918
EP_IN_DMA_ERROR_ADDRESS_REG_2 Register	0x000509B4	Table 902, p. 919
EP_IN_DMA_ERROR_ADDRESS_REG_3 Register	0x000509B8	Table 903, p. 919
EP_IN_DMA_ERROR_ADDRESS_REG_4 Register	0x000509BC	Table 904, p. 919
EP_IN_DMA_ERROR_ADDRESS_REG_5 Register	0x000509C0	Table 905, p. 920
EP_IN_DMA_ERROR_ADDRESS_REG_6 Register	0x000509C4	Table 906, p. 920
EP_IN_DMA_ERROR_ADDRESS_REG_7 Register	0x000509C8	Table 907, p. 920
EP_IN_DMA_ERROR_ADDRESS_REG_8 Register	0x000509CC	Table 908, p. 921
EP_IN_DMA_ERROR_ADDRESS_REG_9 Register	0x000509D0	Table 909, p. 921
EP_IN_DMA_ERROR_ADDRESS_REG_10 Register	0x000509D4	Table 910, p. 921
EP_IN_DMA_ERROR_ADDRESS_REG_11 Register	0x000509D8	Table 911, p. 922
EP_IN_DMA_ERROR_ADDRESS_REG_12 Register	0x000509DC	Table 912, p. 922

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_DMA_ERROR_ADDRESS_REG_13 Register	0x000509E0	Table 913, p. 922
EP_IN_DMA_ERROR_ADDRESS_REG_14 Register	0x000509E4	Table 914, p. 923
EP_IN_DMA_ERROR_ADDRESS_REG_15 Register	0x000509E8	Table 915, p. 923
DMA_SUSPEND_REG Register	0x000509EC	Table 916, p. 923
DMA_SUSPEND_DONE_REG Register	0x000509F0	Table 917, p. 926
DMA_HALT_REG Register	0x000509F4	Table 918, p. 928
DMA_HALT_DONE_REG Register	0x000509F8	Table 919, p. 928
AXI_TIMEOUT_VAL_REG_0 Register	0x000509FC	Table 920, p. 929
AXI_STATE_REG_1 Register	0x00050A00	Table 921, p. 929
AXI_STATE_REG_2 Register	0x00050A04	Table 922, p. 930
AXI_STATE_REG_3 Register	0x00050A08	Table 923, p. 930
SS_SYS_INT_REG Register	0x00050A0C	Table 924, p. 931
SS_SYS_INT_EN_REG Register	0x00050A10	Table 925, p. 934
AXI_ERROR_RESPONSE_CODE_REG_1 Register	0x00050A14	Table 926, p. 935
AXI_ERROR_REPONSE_CODE_REG_2 Register	0x00050A18	Table 927, p. 936
AXI_ERROR_RESPONSE_CODE_REG_3 Register	0x00050A1C	Table 928, p. 936
AXI_ERROR_RESPONSE_CODE_REG_4 Register	0x00050A20	Table 929, p. 937
DMA_STATE_REG_1 Register	0x00050A24	Table 930, p. 937
DMA_STATE_REG_2 Register	0x00050A28	Table 931, p. 937
DMA_STATE_REG_3 Register	0x00050A2C	Table 932, p. 938
DMA_STATE_REG_4 Register	0x00050A30	Table 933, p. 938
DMA_STATE_REG_5 Register	0x00050A34	Table 934, p. 939
SEGMENT_COUNTER_REG_0 Register	0x00050A38	Table 935, p. 939
SEGMENT_COUNTER_REG_1 Register	0x00050A3C	Table 936, p. 939
SEGMENT_COUNTER_REG_2 Register	0x00050A40	Table 937, p. 940
SEGMENT_COUNTER_REG_3 Register	0x00050A44	Table 938, p. 940
SEGMENT_COUNTER_REG_4 Register	0x00050A48	Table 939, p. 940
SEGMENT_COUNTER_REG_5 Register	0x00050A4C	Table 940, p. 941
SEGMENT_COUNTER_REG_6 Register	0x00050A50	Table 941, p. 941
SEGMENT_COUNTER_REG_7 Register	0x00050A54	Table 942, p. 941
SEGMENT_COUNTER_REG_8 Register	0x00050A58	Table 943, p. 942
SEGMENT_COUNTER_REG_9 Register	0x00050A5C	Table 944, p. 942
SEGMENT_COUNTER_REG_10 Register	0x00050A60	Table 945, p. 942
SEGMENT_COUNTER_REG_11 Register	0x00050A64	Table 946, p. 943
SEGMENT_COUNTER_REG_12 Register	0x00050A68	Table 947, p. 943
SEGMENT_COUNTER_REG_13 Register	0x00050A6C	Table 948, p. 943
SEGMENT_COUNTER_REG_14 Register	0x00050A70	Table 949, p. 944
SEGMENT_COUNTER_REG_15 Register	0x00050A74	Table 950, p. 944
EP_IN_DONEQ_START_ADDRESS_REG_0 Register	0x00050A78	Table 951, p. 944
EP_IN_DONEQ_START_ADDRESS_REG_1 Register	0x00050A7C	Table 952, p. 944
EP_IN_DONEQ_START_ADDRESS_REG_2 Register	0x00050A80	Table 953, p. 945

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_DONEQ_START_ADDRESS_REG_3 Register	0x00050A84	Table 954, p. 945
EP_IN_DONEQ_START_ADDRESS_REG_4 Register	0x00050A88	Table 955, p. 945
EP_IN_DONEQ_START_ADDRESS_REG_5 Register	0x00050A8C	Table 956, p. 945
EP_IN_DONEQ_START_ADDRESS_REG_6 Register	0x00050A90	Table 957, p. 945
EP_IN_DONEQ_START_ADDRESS_REG_7 Register	0x00050A94	Table 958, p. 945
EP_IN_DONEQ_START_ADDRESS_REG_8 Register	0x00050A98	Table 959, p. 946
EP_IN_DONEQ_START_ADDRESS_REG_9 Register	0x00050A9C	Table 960, p. 946
EP_IN_DONEQ_START_ADDRESS_REG_10 Register	0x00050AA0	Table 961, p. 946
EP_IN_DONEQ_START_ADDRESS_REG_11 Register	0x00050AA4	Table 962, p. 946
EP_IN_DONEQ_START_ADDRESS_REG_12 Register	0x00050AA8	Table 963, p. 946
EP_IN_DONEQ_START_ADDRESS_REG_13 Register	0x00050AAC	Table 964, p. 946
EP_IN_DONEQ_START_ADDRESS_REG_14 Register	0x00050AB0	Table 965, p. 947
EP_IN_DONEQ_START_ADDRESS_REG_15 Register	0x00050AB4	Table 966, p. 947
EP_IN_DONEQ_END_ADDRESS_REG_0 Register	0x00050AB8	Table 967, p. 947
EP_IN_DONEQ_END_ADDRESS_REG_1 Register	0x00050ABC	Table 968, p. 947
EP_IN_DONEQ_END_ADDRESS_REG_2 Register	0x00050AC0	Table 969, p. 947
EP_IN_DONEQ_END_ADDRESS_REG_3 Register	0x00050AC4	Table 970, p. 947
EP_IN_DONEQ_END_ADDRESS_REG_4 Register	0x00050AC8	Table 971, p. 948
EP_IN_DONEQ_END_ADDRESS_REG_5 Register	0x00050ACC	Table 972, p. 948
EP_IN_DONEQ_END_ADDRESS_REG_6 Register	0x00050AD0	Table 973, p. 948
EP_IN_DONEQ_END_ADDRESS_REG_7 Register	0x00050AD4	Table 974, p. 948
EP_IN_DONEQ_END_ADDRESS_REG_8 Register	0x00050AD8	Table 975, p. 948
EP_IN_DONEQ_END_ADDRESS_REG_9 Register	0x00050ADC	Table 976, p. 948
EP_IN_DONEQ_END_ADDRESS_REG_10 Register	0x00050AE0	Table 977, p. 949
EP_IN_DONEQ_END_ADDRESS_REG_11 Register	0x00050AE4	Table 978, p. 949
EP_IN_DONEQ_END_ADDRESS_REG_12 Register	0x00050AE8	Table 979, p. 949
EP_IN_DONEQ_END_ADDRESS_REG_13 Register	0x00050AEC	Table 980, p. 949
EP_IN_DONEQ_END_ADDRESS_REG_14 Register	0x00050AF0	Table 981, p. 949
EP_IN_DONEQ_END_ADDRESS_REG_15 Register	0x00050AF4	Table 982, p. 949
EP_OUT_DONEQ_START_ADDRESS_REG_0 Register	0x00050AF8	Table 983, p. 950
EP_OUT_DONEQ_START_ADDRESS_REG_1 Register	0x00050AFC	Table 984, p. 950
EP_OUT_DONEQ_START_ADDRESS_REG_2 Register	0x00050B00	Table 985, p. 950
EP_OUT_DONEQ_START_ADDRESS_REG_3 Register	0x00050B04	Table 986, p. 950
EP_OUT_DONEQ_START_ADDRESS_REG_4 Register	0x00050B08	Table 987, p. 950
EP_OUT_DONEQ_START_ADDRESS_REG_5 Register	0x00050B0C	Table 988, p. 950
EP_OUT_DONEQ_START_ADDRESS_REG_6 Register	0x00050B10	Table 989, p. 951
EP_OUT_DONEQ_START_ADDRESS_REG_7 Register	0x00050B14	Table 990, p. 951
EP_OUT_DONEQ_START_ADDRESS_REG_8 Register	0x00050B18	Table 991, p. 951
EP_OUT_DONEQ_START_ADDRESS_REG_9 Register	0x00050B1C	Table 992, p. 951
EP_OUT_DONEQ_START_ADDRESS_REG_10 Register	0x00050B20	Table 993, p. 951

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_OUT_DONEQ_START_ADDRESS_REG_11 Register	0x00050B24	Table 994, p. 951
EP_OUT_DONEQ_START_ADDRESS_REG_12 Register	0x00050B28	Table 995, p. 952
EP_OUT_DONEQ_START_ADDRESS_REG_13 Register	0x00050B2C	Table 996, p. 952
EP_OUT_DONEQ_START_ADDRESS_REG_14 Register	0x00050B30	Table 997, p. 952
EP_OUT_DONEQ_START_ADDRESS_REG_15 Register	0x00050B34	Table 998, p. 952
EP_OUT_DONEQ_END_ADDRESS_REG_0 Register	0x00050B38	Table 999, p. 952
EP_OUT_DONEQ_END_ADDRESS_REG_1 Register	0x00050B3C	Table 1000, p. 952
EP_OUT_DONEQ_END_ADDRESS_REG_2 Register	0x00050B40	Table 1001, p. 953
EP_OUT_DONEQ_END_ADDRESS_REG_3 Register	0x00050B44	Table 1002, p. 953
EP_OUT_DONEQ_END_ADDRESS_REG_4 Register	0x00050B48	Table 1003, p. 953
EP_OUT_DONEQ_END_ADDRESS_REG_5 Register	0x00050B4C	Table 1004, p. 953
EP_OUT_DONEQ_END_ADDRESS_REG_6 Register	0x00050B50	Table 1005, p. 953
EP_OUT_DONEQ_END_ADDRESS_REG_7 Register	0x00050B54	Table 1006, p. 953
EP_OUT_DONEQ_END_ADDRESS_REG_8 Register	0x00050B58	Table 1007, p. 954
EP_OUT_DONEQ_END_ADDRESS_REG_9 Register	0x00050B5C	Table 1008, p. 954
EP_OUT_DONEQ_END_ADDRESS_REG_10 Register	0x00050B60	Table 1009, p. 954
EP_OUT_DONEQ_END_ADDRESS_REG_11 Register	0x00050B64	Table 1010, p. 954
EP_OUT_DONEQ_END_ADDRESS_REG_12 Register	0x00050B68	Table 1011, p. 954
EP_OUT_DONEQ_END_ADDRESS_REG_13 Register	0x00050B6C	Table 1012, p. 954
EP_OUT_DONEQ_END_ADDRESS_REG_14 Register	0x00050B70	Table 1013, p. 955
EP_OUT_DONEQ_END_ADDRESS_REG_15 Register	0x00050B74	Table 1014, p. 955
EP_IN_DONEQ_WRITE_POINTER_REG_0 Register	0x00050B78	Table 1015, p. 955
EP_IN_DONEQ_WRITE_POINTER_REG_1 Register	0x00050B7C	Table 1016, p. 955
EP_IN_DONEQ_WRITE_POINTER_REG_2 Register	0x00050B80	Table 1017, p. 955
EP_IN_DONEQ_WRITE_POINTER_REG_3 Register	0x00050B84	Table 1018, p. 956
EP_IN_DONEQ_WRITE_POINTER_REG_4 Register	0x00050B88	Table 1019, p. 956
EP_IN_DONEQ_WRITE_POINTER_REG_5 Register	0x00050B8C	Table 1020, p. 956
EP_IN_DONEQ_WRITE_POINTER_REG_6 Register	0x00050B90	Table 1021, p. 956
EP_IN_DONEQ_WRITE_POINTER_REG_7 Register	0x00050B94	Table 1022, p. 956
EP_IN_DONEQ_WRITE_POINTER_REG_8 Register	0x00050B98	Table 1023, p. 957
EP_IN_DONEQ_WRITE_POINTER_REG_9 Register	0x00050B9C	Table 1024, p. 957
EP_IN_DONEQ_WRITE_POINTER_REG_10 Register	0x00050BA0	Table 1025, p. 957
EP_IN_DONEQ_WRITE_POINTER_REG_11 Register	0x00050BA4	Table 1026, p. 957
EP_IN_DONEQ_WRITE_POINTER_REG_12 Register	0x00050BA8	Table 1027, p. 957
EP_IN_DONEQ_WRITE_POINTER_REG_13 Register	0x00050BAC	Table 1028, p. 958
EP_IN_DONEQ_WRITE_POINTER_REG_14 Register	0x00050BB0	Table 1029, p. 958
EP_IN_DONEQ_WRITE_POINTER_REG_15 Register	0x00050BB4	Table 1030, p. 958
EP_IN_DONEQ_READ_POINTER_REG_0 Register	0x00050BB8	Table 1031, p. 958

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_DONEQ_READ_POINTER_REG_1 Register	0x00050BBC	Table 1032, p. 958
EP_IN_DONEQ_READ_POINTER_REG_2 Register	0x00050BC0	Table 1033, p. 959
EP_IN_DONEQ_READ_POINTER_REG_3 Register	0x00050BC4	Table 1034, p. 959
EP_IN_DONEQ_READ_POINTER_REG_4 Register	0x00050BC8	Table 1035, p. 959
EP_IN_DONEQ_READ_POINTER_REG_5 Register	0x00050BCC	Table 1036, p. 959
EP_IN_DONEQ_READ_POINTER_REG_6 Register	0x00050BD0	Table 1037, p. 960
EP_IN_DONEQ_READ_POINTER_REG_7 Register	0x00050BD4	Table 1038, p. 960
EP_IN_DONEQ_READ_POINTER_REG_8 Register	0x00050BD8	Table 1039, p. 960
EP_IN_DONEQ_READ_POINTER_REG_9 Register	0x00050BDC	Table 1040, p. 960
EP_IN_DONEQ_READ_POINTER_REG_10 Register	0x00050BE0	Table 1041, p. 961
EP_IN_DONEQ_READ_POINTER_REG_11 Register	0x00050BE4	Table 1042, p. 961
EP_IN_DONEQ_READ_POINTER_REG_12 Register	0x00050BE8	Table 1043, p. 961
EP_IN_DONEQ_READ_POINTER_REG_13 Register	0x00050BEC	Table 1044, p. 961
EP_IN_DONEQ_READ_POINTER_REG_14 Register	0x00050BF0	Table 1045, p. 962
EP_IN_DONEQ_READ_POINTER_REG_15 Register	0x00050BF4	Table 1046, p. 962
EP_OUT_DONEQ_WRITE_POINTER_REG_0 Register	0x00050BF8	Table 1047, p. 962
EP_OUT_DONEQ_WRITE_POINTER_REG_1 Register	0x00050BFC	Table 1048, p. 962
EP_OUT_DONEQ_WRITE_POINTER_REG_2 Register	0x00050C00	Table 1049, p. 962
EP_OUT_DONEQ_WRITE_POINTER_REG_3 Register	0x00050C04	Table 1050, p. 963
EP_OUT_DONEQ_WRITE_POINTER_REG_4 Register	0x00050C08	Table 1051, p. 963
EP_OUT_DONEQ_WRITE_POINTER_REG_5 Register	0x00050C0C	Table 1052, p. 963
EP_OUT_DONEQ_WRITE_POINTER_REG_6 Register	0x00050C10	Table 1053, p. 963
EP_OUT_DONEQ_WRITE_POINTER_REG_7 Register	0x00050C14	Table 1054, p. 963
EP_OUT_DONEQ_WRITE_POINTER_REG_8 Register	0x00050C18	Table 1055, p. 964
EP_OUT_DONEQ_WRITE_POINTER_REG_9 Register	0x00050C1C	Table 1056, p. 964
EP_OUT_DONEQ_WRITE_POINTER_REG_10 Register	0x00050C20	Table 1057, p. 964
EP_OUT_DONEQ_WRITE_POINTER_REG_11 Register	0x00050C24	Table 1058, p. 964
EP_OUT_DONEQ_WRITE_POINTER_REG_12 Register	0x00050C28	Table 1059, p. 964
EP_OUT_DONEQ_WRITE_POINTER_REG_13 Register	0x00050C2C	Table 1060, p. 965
EP_OUT_DONEQ_WRITE_POINTER_REG_14 Register	0x00050C30	Table 1061, p. 965
EP_OUT_DONEQ_WRITE_POINTER_REG_15 Register	0x00050C34	Table 1062, p. 965
EP_OUT_DONEQ_READ_POINTER_REG_0 Register	0x00050C38	Table 1063, p. 965
EP_OUT_DONEQ_READ_POINTER_REG_1 Register	0x00050C3C	Table 1064, p. 965
EP_OUT_DONEQ_READ_POINTER_REG_2 Register	0x00050C40	Table 1065, p. 966
EP_OUT_DONEQ_READ_POINTER_REG_3 Register	0x00050C44	Table 1066, p. 966
EP_OUT_DONEQ_READ_POINTER_REG_4 Register	0x00050C48	Table 1067, p. 966
EP_OUT_DONEQ_READ_POINTER_REG_5 Register	0x00050C4C	Table 1068, p. 966
EP_OUT_DONEQ_READ_POINTER_REG_6 Register	0x00050C50	Table 1069, p. 967
EP_OUT_DONEQ_READ_POINTER_REG_7 Register	0x00050C54	Table 1070, p. 967
EP_OUT_DONEQ_READ_POINTER_REG_8 Register	0x00050C58	Table 1071, p. 967
EP_OUT_DONEQ_READ_POINTER_REG_9 Register	0x00050C5C	Table 1072, p. 967

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_OUT_DONEQ_READ_POINTER_REG_10 Register	0x00050C60	Table 1073, p. 968
EP_OUT_DONEQ_READ_POINTER_REG_11 Register	0x00050C64	Table 1074, p. 968
EP_OUT_DONEQ_READ_POINTER_REG_12 Register	0x00050C68	Table 1075, p. 968
EP_OUT_DONEQ_READ_POINTER_REG_13 Register	0x00050C6C	Table 1076, p. 968
EP_OUT_DONEQ_READ_POINTER_REG_14 Register	0x00050C70	Table 1077, p. 969
EP_OUT_DONEQ_READ_POINTER_REG_15 Register	0x00050C74	Table 1078, p. 969
DONEQ_FULL_STATUS_REG Register	0x00050C78	Table 1079, p. 969
ABORT_REQ_STATUS_REG Register	0x00050C7C	Table 1080, p. 974
ABORT_ID_REG_1 Register	0x00050C84	Table 1081, p. 977
ABORT_ID_REG_2 Register	0x00050C88	Table 1082, p. 978
ABORT_ID_REG_3 Register	0x00050C8C	Table 1083, p. 978
ABORT_ID_REG_4 Register	0x00050C90	Table 1084, p. 978
ABORT_ID_REG_5 Register	0x00050C94	Table 1085, p. 979
ABORT_ID_REG_6 Register	0x00050C98	Table 1086, p. 979
ABORT_ID_REG_7 Register	0x00050C9C	Table 1087, p. 980
ABORT_ID_REG_8 Register	0x00050CA0	Table 1088, p. 980
ABORT_ID_REG_9 Register	0x00050CA4	Table 1089, p. 980
ABORT_ID_REG_10 Register	0x00050CA8	Table 1090, p. 981
ABORT_ID_REG_11 Register	0x00050CAC	Table 1091, p. 981
ABORT_ID_REG_12 Register	0x00050CB0	Table 1092, p. 982
ABORT_ID_REG_13 Register	0x00050CB4	Table 1093, p. 982
ABORT_ID_REG_14 Register	0x00050CB8	Table 1094, p. 982
ABORT_ID_REG_15 Register	0x00050CBC	Table 1095, p. 983
FLUSHED_BD_COUNTER_REG_1 Register	0x00050CC0	Table 1096, p. 983
FLUSHED_BD_COUNTER_REG_2 Register	0x00050CC4	Table 1097, p. 984
FLUSHED_BD_COUNTER_REG_3 Register	0x00050CC8	Table 1098, p. 984
FLUSHED_BD_COUNTER_REG_4 Register	0x00050CCC	Table 1099, p. 984
FLUSHED_BD_COUNTER_REG_5 Register	0x00050CD0	Table 1100, p. 985
FLUSHED_BD_COUNTER_REG_6 Register	0x00050CD4	Table 1101, p. 985
FLUSHED_BD_COUNTER_REG_7 Register	0x00050CD8	Table 1102, p. 985
FLUSHED_BD_COUNTER_REG_8 Register	0x00050CDC	Table 1103, p. 986
FLUSHED_BD_COUNTER_REG_9 Register	0x00050CE0	Table 1104, p. 986
FLUSHED_BD_COUNTER_REG_10 Register	0x00050CE4	Table 1105, p. 986
FLUSHED_BD_COUNTER_REG_11 Register	0x00050CE8	Table 1106, p. 987
FLUSHED_BD_COUNTER_REG_12 Register	0x00050CEC	Table 1107, p. 987
FLUSHED_BD_COUNTER_REG_13 Register	0x00050CF0	Table 1108, p. 987
FLUSHED_BD_COUNTER_REG_14 Register	0x00050CF4	Table 1109, p. 988
FLUSHED_BD_COUNTER_REG_15 Register	0x00050CF8	Table 1110, p. 988
EP_IN_FLUSHED_SID_BITMAP_REG_1 Register	0x00050CFC	Table 1111, p. 988
EP_IN_FLUSHED_SID_BITMAP_REG_2 Register	0x00050D1C	Table 1112, p. 988
EP_IN_FLUSHED_SID_BITMAP_REG_3 Register	0x00050D3C	Table 1113, p. 989

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_FLUSHED_SID_BITMAP_REG_4 Register	0x00050D5C	Table 1114, p. 989
EP_IN_FLUSHED_SID_BITMAP_REG_5 Register	0x00050D7C	Table 1115, p. 989
EP_IN_FLUSHED_SID_BITMAP_REG_6 Register	0x00050D9C	Table 1116, p. 989
EP_IN_FLUSHED_SID_BITMAP_REG_7 Register	0x00050DBC	Table 1117, p. 989
EP_IN_FLUSHED_SID_BITMAP_REG_8 Register	0x00050DDC	Table 1118, p. 990
EP_IN_FLUSHED_SID_BITMAP_REG_9 Register	0x00050DFC	Table 1119, p. 990
EP_IN_FLUSHED_SID_BITMAP_REG_10 Register	0x00050E1C	Table 1120, p. 990
EP_IN_FLUSHED_SID_BITMAP_REG_11 Register	0x00050E3C	Table 1121, p. 990
EP_IN_FLUSHED_SID_BITMAP_REG_12 Register	0x00050E5C	Table 1122, p. 990
EP_IN_FLUSHED_SID_BITMAP_REG_13 Register	0x00050E7C	Table 1123, p. 991
EP_IN_FLUSHED_SID_BITMAP_REG_14 Register	0x00050E9C	Table 1124, p. 991
EP_IN_FLUSHED_SID_BITMAP_REG_15 Register	0x00050EBC	Table 1125, p. 991
EP_OUT_FLUSHED_SID_BITMAP_REG_1 Register	0x00050EDC	Table 1126, p. 991
EP_OUT_FLUSHED_SID_BITMAP_REG_2 Register	0x00050EFC	Table 1127, p. 991
EP_OUT_FLUSHED_SID_BITMAP_REG_3 Register	0x00050F1C	Table 1128, p. 992
EP_OUT_FLUSHED_SID_BITMAP_REG_4 Register	0x00050F3C	Table 1129, p. 992
EP_OUT_FLUSHED_SID_BITMAP_REG_5 Register	0x00050F5C	Table 1130, p. 992
EP_OUT_FLUSHED_SID_BITMAP_REG_6 Register	0x00050F7C	Table 1131, p. 992
EP_OUT_FLUSHED_SID_BITMAP_REG_7 Register	0x00050F9C	Table 1132, p. 992
EP_OUT_FLUSHED_SID_BITMAP_REG_8 Register	0x00050FBC	Table 1133, p. 993
EP_OUT_FLUSHED_SID_BITMAP_REG_9 Register	0x00050FDC	Table 1134, p. 993
EP_OUT_FLUSHED_SID_BITMAP_REG_10 Register	0x00050FFC	Table 1135, p. 993
EP_OUT_FLUSHED_SID_BITMAP_REG_11 Register	0x0005101C	Table 1136, p. 993
EP_OUT_FLUSHED_SID_BITMAP_REG_12 Register	0x0005103C	Table 1137, p. 993
EP_OUT_FLUSHED_SID_BITMAP_REG_13 Register	0x0005105C	Table 1138, p. 994
EP_OUT_FLUSHED_SID_BITMAP_REG_14 Register	0x0005107C	Table 1139, p. 994
EP_OUT_FLUSHED_SID_BITMAP_REG_15 Register	0x0005109C	Table 1140, p. 994
EP_IN_NRDYQ_HEAD_POINTER_REG_1 Register	0x000510BC	Table 1141, p. 994
EP_IN_NRDYQ_HEAD_POINTER_REG_2 Register	0x000510C0	Table 1142, p. 994
EP_IN_NRDYQ_HEAD_POINTER_REG_3 Register	0x000510C4	Table 1143, p. 995
EP_IN_NRDYQ_HEAD_POINTER_REG_4 Register	0x000510C8	Table 1144, p. 995
EP_IN_NRDYQ_HEAD_POINTER_REG_5 Register	0x000510CC	Table 1145, p. 995
EP_IN_NRDYQ_HEAD_POINTER_REG_6 Register	0x000510D0	Table 1146, p. 995
EP_IN_NRDYQ_HEAD_POINTER_REG_7 Register	0x000510D4	Table 1147, p. 995
EP_IN_NRDYQ_HEAD_POINTER_REG_8 Register	0x000510D8	Table 1148, p. 996
EP_IN_NRDYQ_HEAD_POINTER_REG_9 Register	0x000510DC	Table 1149, p. 996
EP_IN_NRDYQ_HEAD_POINTER_REG_10 Register	0x000510E0	Table 1150, p. 996
EP_IN_NRDYQ_HEAD_POINTER_REG_11 Register	0x000510E4	Table 1151, p. 996
EP_IN_NRDYQ_HEAD_POINTER_REG_12 Register	0x000510E8	Table 1152, p. 996
EP_IN_NRDYQ_HEAD_POINTER_REG_13 Register	0x000510EC	Table 1153, p. 997
EP_IN_NRDYQ_HEAD_POINTER_REG_14 Register	0x000510F0	Table 1154, p. 997

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_IN_NRDYQ_HEAD_POINTER_REG_15 Register	0x000510F4	Table 1155, p. 997
EP_OUT_NRDYQ_HEAD_POINTER_REG_1 Register	0x000510F8	Table 1156, p. 997
EP_OUT_NRDYQ_HEAD_POINTER_REG_2 Register	0x000510FC	Table 1157, p. 997
EP_OUT_NRDYQ_HEAD_POINTER_REG_3 Register	0x00051100	Table 1158, p. 998
EP_OUT_NRDYQ_HEAD_POINTER_REG_4 Register	0x00051104	Table 1159, p. 998
EP_OUT_NRDYQ_HEAD_POINTER_REG_5 Register	0x00051108	Table 1160, p. 998
EP_OUT_NRDYQ_HEAD_POINTER_REG_6 Register	0x0005110C	Table 1161, p. 998
EP_OUT_NRDYQ_HEAD_POINTER_REG_7 Register	0x00051110	Table 1162, p. 998
EP_OUT_NRDYQ_HEAD_POINTER_REG_8 Register	0x00051114	Table 1163, p. 999
EP_OUT_NRDYQ_HEAD_POINTER_REG_9 Register	0x00051118	Table 1164, p. 999
EP_OUT_NRDYQ_HEAD_POINTER_REG_10 Register	0x0005111C	Table 1165, p. 999
EP_OUT_NRDYQ_HEAD_POINTER_REG_11 Register	0x00051120	Table 1166, p. 999
EP_OUT_NRDYQ_HEAD_POINTER_REG_12 Register	0x00051124	Table 1167, p. 999
EP_OUT_NRDYQ_HEAD_POINTER_REG_13 Register	0x00051128	Table 1168, p. 1000
EP_OUT_NRDYQ_HEAD_POINTER_REG_14 Register	0x0005112C	Table 1169, p. 1000
EP_OUT_NRDYQ_HEAD_POINTER_REG_15 Register	0x00051130	Table 1170, p. 1000
EP_IN_NRDYQ_TAIL_POINTER_REG_1 Register	0x00051134	Table 1171, p. 1000
EP_IN_NRDYQ_TAIL_POINTER_REG_2 Register	0x00051138	Table 1172, p. 1000
EP_IN_NRDYQ_TAIL_POINTER_REG_3 Register	0x0005113C	Table 1173, p. 1001
EP_IN_NRDYQ_TAIL_POINTER_REG_4 Register	0x00051140	Table 1174, p. 1001
EP_IN_NRDYQ_TAIL_POINTER_REG_5 Register	0x00051144	Table 1175, p. 1001
EP_IN_NRDYQ_TAIL_POINTER_REG_6 Register	0x00051148	Table 1176, p. 1001
EP_IN_NRDYQ_TAIL_POINTER_REG_7 Register	0x0005114C	Table 1177, p. 1001
EP_IN_NRDYQ_TAIL_POINTER_REG_8 Register	0x00051150	Table 1178, p. 1002
EP_IN_NRDYQ_TAIL_POINTER_REG_9 Register	0x00051154	Table 1179, p. 1002
EP_IN_NRDYQ_TAIL_POINTER_REG_10 Register	0x00051158	Table 1180, p. 1002
EP_IN_NRDYQ_TAIL_POINTER_REG_11 Register	0x0005115C	Table 1181, p. 1002
EP_IN_NRDYQ_TAIL_POINTER_REG_12 Register	0x00051160	Table 1182, p. 1002
EP_IN_NRDYQ_TAIL_POINTER_REG_13 Register	0x00051164	Table 1183, p. 1003
EP_IN_NRDYQ_TAIL_POINTER_REG_14 Register	0x00051168	Table 1184, p. 1003
EP_IN_NRDYQ_TAIL_POINTER_REG_15 Register	0x0005116C	Table 1185, p. 1003
EP_OUT_NRDYQ_TAIL_POINTER_REG_1 Register	0x00051170	Table 1186, p. 1003
EP_OUT_NRDYQ_TAIL_POINTER_REG_2 Register	0x00051174	Table 1187, p. 1003
EP_OUT_NRDYQ_TAIL_POINTER_REG_3 Register	0x00051178	Table 1188, p. 1004
EP_OUT_NRDYQ_TAIL_POINTER_REG_4 Register	0x0005117C	Table 1189, p. 1004
EP_OUT_NRDYQ_TAIL_POINTER_REG_5 Register	0x00051180	Table 1190, p. 1004
EP_OUT_NRDYQ_TAIL_POINTER_REG_6 Register	0x00051184	Table 1191, p. 1004
EP_OUT_NRDYQ_TAIL_POINTER_REG_7 Register	0x00051188	Table 1192, p. 1004
EP_OUT_NRDYQ_TAIL_POINTER_REG_8 Register	0x0005118C	Table 1193, p. 1005
EP_OUT_NRDYQ_TAIL_POINTER_REG_9 Register	0x00051190	Table 1194, p. 1005
EP_OUT_NRDYQ_TAIL_POINTER_REG_10 Register	0x00051194	Table 1195, p. 1005

Copyright © 2015 Marvell

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
EP_OUT_NRDYQ_TAIL_POINTER_REG_11 Register	0x00051198	Table 1196, p. 1005
EP_OUT_NRDYQ_TAIL_POINTER_REG_12 Register	0x0005119C	Table 1197, p. 1005
EP_OUT_NRDYQ_TAIL_POINTER_REG_13 Register	0x000511A0	Table 1198, p. 1006
EP_OUT_NRDYQ_TAIL_POINTER_REG_14 Register	0x000511A4	Table 1199, p. 1006
EP_OUT_NRDYQ_TAIL_POINTER_REG_15 Register	0x000511A8	Table 1200, p. 1006
EP_IN_RDYQ_HEAD_POINTER_REG_1 Register	0x000511AC	Table 1201, p. 1006
EP_IN_RDYQ_HEAD_POINTER_REG_2 Register	0x000511B0	Table 1202, p. 1006
EP_IN_RDYQ_HEAD_POINTER_REG_3 Register	0x000511B4	Table 1203, p. 1007
EP_IN_RDYQ_HEAD_POINTER_REG_4 Register	0x000511B8	Table 1204, p. 1007
EP_IN_RDYQ_HEAD_POINTER_REG_5 Register	0x000511BC	Table 1205, p. 1007
EP_IN_RDYQ_HEAD_POINTER_REG_6 Register	0x000511C0	Table 1206, p. 1007
EP_IN_RDYQ_HEAD_POINTER_REG_7 Register	0x000511C4	Table 1207, p. 1007
EP_IN_RDYQ_HEAD_POINTER_REG_8 Register	0x000511C8	Table 1208, p. 1008
EP_IN_RDYQ_HEAD_POINTER_REG_9 Register	0x000511CC	Table 1209, p. 1008
EP_IN_RDYQ_HEAD_POINTER_REG_10 Register	0x000511D0	Table 1210, p. 1008
EP_IN_RDYQ_HEAD_POINTER_REG_11 Register	0x000511D4	Table 1211, p. 1008
EP_IN_RDYQ_HEAD_POINTER_REG_12 Register	0x000511D8	Table 1212, p. 1008
EP_IN_RDYQ_HEAD_POINTER_REG_13 Register	0x000511DC	Table 1213, p. 1009
EP_IN_RDYQ_HEAD_POINTER_REG_14 Register	0x000511E0	Table 1214, p. 1009
EP_IN_RDYQ_HEAD_POINTER_REG_15 Register	0x000511E4	Table 1215, p. 1009
EP_OUT_RDYQ_HEAD_POINTER_REG_1 Register	0x000511E8	Table 1216, p. 1009
EP_OUT_RDYQ_HEAD_POINTER_REG_2 Register	0x000511EC	Table 1217, p. 1009
EP_OUT_RDYQ_HEAD_POINTER_REG_3 Register	0x000511F0	Table 1218, p. 1010
EP_OUT_RDYQ_HEAD_POINTER_REG_4 Register	0x000511F4	Table 1219, p. 1010
EP_OUT_RDYQ_HEAD_POINTER_REG_5 Register	0x000511F8	Table 1220, p. 1010
EP_OUT_RDYQ_HEAD_POINTER_REG_6 Register	0x000511FC	Table 1221, p. 1010
EP_OUT_RDYQ_HEAD_POINTER_REG_7 Register	0x00051200	Table 1222, p. 1010
EP_OUT_RDYQ_HEAD_POINTER_REG_8 Register	0x00051204	Table 1223, p. 1011
EP_OUT_RDYQ_HEAD_POINTER_REG_9 Register	0x00051208	Table 1224, p. 1011
EP_OUT_RDYQ_HEAD_POINTER_REG_10 Register	0x0005120C	Table 1225, p. 1011
EP_OUT_RDYQ_HEAD_POINTER_REG_11 Register	0x00051210	Table 1226, p. 1011
EP_OUT_RDYQ_HEAD_POINTER_REG_12 Register	0x00051214	Table 1227, p. 1011
EP_OUT_RDYQ_HEAD_POINTER_REG_13 Register	0x00051218	Table 1228, p. 1012
EP_OUT_RDYQ_HEAD_POINTER_REG_14 Register	0x0005121C	Table 1229, p. 1012
EP_OUT_RDYQ_HEAD_POINTER_REG_15 Register	0x00051220	Table 1230, p. 1012
BULK_STREAMING_INT_STATUS_REG Register	0x00051224	Table 1231, p. 1012
BULK_STREAMING_INT_ENABLE_REG Register	0x00051228	Table 1232, p. 1013
PRIME_RCV_STATUS_REG Register	0x0005122C	Table 1233, p. 1013
HIMD_RCV_STATUS_REG Register	0x00051230	Table 1234, p. 1015
STREAM_REJ_STATUS_REG Register	0x00051234	Table 1235, p. 1016
HOST_FLOWCTRL_STATUS_REG Register	0x00051238	Table 1236, p. 1018

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 650: Summary Map Table for the USB 3.0 Device Registers (Continued)

Register Name	Offset	Table and Page
BD_MEM_IN_REG_1 Register	0x0005123C	Table 1237, p. 1020
PL_DEBUG_REG_1 Register	0x00051264	Table 1238, p. 1020
PL_DEBUG_REG_2 Register	0x00051268	Table 1239, p. 1020
PL_DEBUG_REG_3 Register	0x0005126C	Table 1240, p. 1021
PL_DEBUG_REG_4 Register	0x00051270	Table 1241, p. 1021
DMA_DBG_IN_REG_1 Register	0x00051274	Table 1242, p. 1021
DMA_DBG_IN_REG_2 Register	0x00051278	Table 1243, p. 1021
DMA_DBG_OUT_REG_1 Register	0x0005127C	Table 1244, p. 1022
DMA_DBG_OUT_REG_2 Register	0x00051280	Table 1245, p. 1022
USB 2 Modified		
USB2_USBSTS Register	0x00054144	Table 1246, p. 1022
USB2_USBINT Register	0x00054148	Table 1247, p. 1023
USB2_DEVLC Register	0x0005415C	Table 1248, p. 1024
USB2_PORTSC Register	0x00054184	Table 1249, p. 1025

A.13.1 USB 3.0 Device Controller

A.13.1.1 MAC

Table 651: TSEQ_COUNTER_U1U2_TIMEOUT_REG Register
Offset: 0x00050190

Bit	Field	Type/InitVal	Description
Internal use register			
31:16	usb3_ltssm_rxeq_cc_val	RW 0xFFFF	usb3_ltssm_rxeq_cc_val max value for rxeq_counter
15	Reserved_1	RO 0x0	Reserved_1
14:8	usb3_u2_exit_tto_val	RW 0x54	usb3_u2_exit_tto_val u2 exit timeout value, 100us per unit.
7	Reserved_2	RO 0x0	Reserved_2
6:0	usb3_u1_exit_tto_val	RW 0x54	usb3_u1_exit_tto_val u1 exit timeout value, 100us per unit.

Copyright © 2015 Marvell

Table 652: TIMER_TIMEOUT_REG_1 Register
Offset: 0x00050194

Bit	Field	Type/InitVal	Description
Internal use register			
31:29	Reserved_1	RO 0x0	Reserved_1
28:24	usb3_ltssm_2ms_tto_val	RW 0x1C	usb3_ltssm_2ms_tto_val timeout value for 2ms timer, 1 unit = 100us.
23	Reserved_2	RO 0x0	Reserved_2
22:16	usb3_ltssm_6ms_tto_val	RW 0x54	usb3_ltssm_6ms_tto_val timeout value for 6ms timer, 1 unit = 100us.
15:8	usb3_ltssm_12ms_tto_val	RW 0xA8	usb3_ltssm_12ms_tto_val timeout value for 12ms timer, 1 unit = 100us.
7:6	Reserved_3	RO 0x0	Reserved_3
5:0	usb3_ltssm_100ms_tto_val	RW 0xE	usb3_ltssm_100ms_tto_val timeout value for 100ms timer, 1 unit = 10ms.

Table 653: TIMER_TIMEOUT_REG_2 Register
Offset: 0x00050198

Bit	Field	Type/InitVal	Description
Internal use register			
31:8	Reserved_1	RO 0x0	Reserved_1
7:0	usb3_ltssm_360ms_tto_val	RW 0x32	usb3_ltssm_360ms_tto_val timeout value for 360ms timer, 1 unit = 10ms.

Table 654: LFPS_TX_CONFIG_REG Register
Offset: 0x0005019C

Bit	Field	Type/InitVal	Description
Internal use register			
31:28	Reserved_1	RO 0x0	Reserved_1
27:24	lfps_tx_trepeat_polling	RW 0xB	lfps_tx_trepeat_polling Indicates trepeat duration for polling.lfps transmission (1unit = 1us)
23:16	lfps_tx_trepeat_ping	RW 0xEE	lfps_tx_trepeat_ping Indicates trepeat duration for ping.lfps transmission (1unit = 1ms)
15:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 654: LFPS_TX_CONFIG_REG Register (Continued)
Offset: 0x0005019C

Bit	Field	Type/InitVal	Description
11:8	lfps_tx_tburst_polling	RW 0xC	lfps_tx_tburst_polling Indicates tburst duration for polling.lfps transmission (1unit = 100ns)
7:4	Reserved_3	RO 0x0	Reserved_3
3:0	lfps_tx_tburst_ping	RW 0x3	lfps_tx_tburst_ping Indicates tburst duration for ping.lfps transmission (1unit = 1 ref_clk cycle)

Table 655: LFPS_RX_CONFIG_REG Register
Offset: 0x000501A0

Bit	Field	Type/InitVal	Description
Internal use register			
31:27	Reserved_1	RO 0x0	Reserved_1
26:24	lfps_tx_trepeat_polling	RW 0x4	lfps_rx_trepeat_polling Indicates trepeat duration for polling.lfps transmission (1unit = 1us)
23:16	lfps_rx_trepeat_ping	RW 0x9F	lfps_rx_trepeat_ping Indicates trepeat duration for ping.lfps transmission (1unit = 1ms)
15:12	Reserved_2	RO 0x0	Reserved_2
11:8	lfps_rx_tburst_polling	RW 0x5	lfps_rx_tburst_polling Indicates tburst duration for polling.lfps detection (1unit = 100ns)
7:0	Reserved_3	RO 0x0	Reserved_3

Table 656: LFPS_WR_TRESET_VAL_REG Register
Offset: 0x000501A4

Bit	Field	Type/InitVal	Description
Internal use register			
31:15	Reserved_1	RO 0x0	Reserved_1
14:8	lfps_wr_treset_total	RW 0x76	lfps_wr_treset_total Indicates trepeat duration for warm reset (1unit = 1ms)
7	Reserved_2	RO 0x0	Reserved_2
6:0	lfps_wr_tresetdelay_min	RW 0x12	lfps_wr_tresetdelay_min Indicates warm reset tresetdelay minimum duration (1unit = 1ms)

Copyright © 2015 Marvell

Table 657: COUNTER_DELAY_TX_REG Register
Offset: 0x000501A8

Bit	Field	Type/InitVal	Description
Internal use register			
31:16	cntr_delay_tx_Polling_LFPS	RW 0x0	cntr_delay_tx_Polling_LFPS delay counter for sending polling_LFPS (1unit = 100us)
15:0	cntr_delay_tx_Ping_LFPS	RW 0x0	cntr_delay_tx_Ping_LFPS delay counter for sending ping_LFPS (1unit = 1us)

Table 658: PHY_CONFIG_REG Register
Offset: 0x000501AC

Bit	Field	Type/InitVal	Description
31:16	polling_lfps_send_num_aft_recv_one	RW 0x8	polling_lfps_send_num_aft_recv_one Send this number of lfps after recv one lfps from host. Internal use.
15:6	Reserved	RO 0x0	Reserved
5	tx_swing	RW 0x0	tx_swing controls value of tx_swing pins to SuperSpeed PHY. 0: Full Swing 1: Low Swing Please refer to USB3 PHY specification for the usage model.
4:3	tx_deemph	RW 0x1	tx_deemph controls value of tx_deemph pins to SuperSpeed PHY. 0: -6dB de-emphasis 1: -3.5dB de-emphasis 2: No de-emphasis 3: Reserved Please rrefer to USB3 PHY Spec for how to use this parameter.
2:0	tx_margin	RW 0x0	tx_margin controls value of tx_margin pins to SuperSpeed PHY. Please refer to USB3 PHY Spec for how to use this parameter.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 659: LOOP_BACK_REG Register
Offset: 0x000501B0

Bit	Field	Type/InitVal	Description
31:20	Reserved	RO 0x0	Reserved
19	reg_loopback_mode	RW 0x0	reg_loopback_mode reg_loopback_mode. To have device controller behave like a loopback master (can be used in ATE production test): Setting this bit to 1 before soft_connect register is set.
18	reg_go_loopback	RW 0x0	reg_go_loopback reg_go_loopback. After setting reg_loopback_mode to 1, when SW asserts this bit, device controller will behave like a loopback master. When SW Set this bit to 0, controller will exit loopback to Rx.Detect.
17	reg_ecnt_valid	RO 0x0	reg_ecnt_valid reg_ecnt_valid. In loopback mode when reg_loopback_mode and reg_go_loopback are set to 1, SW should poll this bit to make sure it is set before it reads error_counter field reg_ecnt.
16	reg_lb_ecnt_valid	RO 0x0	reg_lb_ecnt_valid reg_lb_ecnt_valid. In loopback mode when reg_loopback_mode and reg_go_loopback are set to1, SW should poll this bit to make sure it is set before it reads loopback error counter reg_lb_ecnt.
15:8	reg_ecnt	RO 0x0	reg_ecnt The field indicates the error received during loopback. This error is generated when HW compares sneding data with received data.
7:0	reg_lb_ecnt	RO 0x0	reg_lb_ecnt This field indicates the error count recevied from loopback slave.

Table 660: BERT_INTERVAL_REG Register
Offset: 0x000501B4

Bit	Field	Type/InitVal	Description
31:0	reg_bert_interval	RW 0x1400	reg_bert_interval reg_bert_interval. Loopback master sends out BERC with BRST ordered set when "byte interval" is reached. byte interval = reg_bert_interval*128bytes. Default value 32'h1400 means 640k bytes, corresponding to 1ms.

Copyright © 2015 Marvell

Table 661: SHALLOW_LOOP_BACK_REG Register
Offset: 0x000501B8

Bit	Field	Type/InitVal	Description
31:24	shallow_lb_pattern	RW 0xAA	shallow_lb_pattern Before asserting shallow_lb_en, program this field for the shallow loopback pattern desired. Sequences of shallow_lb_pattern (default 0xAA) and its negative (default 0x55) will be continously send to PHY for shallow loopback. Mac will report the comparison result in shallow_lb_ecnt. Can be used on ATE for BIST.
23:17	reserved_1	RO 0x0	reserved_1
16	shallow_lb_en	RW 0x0	shallow_lb_en Set this bit to 1 (without set soft_connect register bit", device will send pattern for supporting Marvell PHY shallow loop-back. Error count will be returned from shallow_lb_ecnt field. Can be used on ATE for BIST.
15:8	reserved_2	RO 0x0	reserved_2
7:0	shallow_lb_ecnt	RO 0x0	shallow_lb_ecnt Error count from mac to Marvell PHY shallow loop-back. If error count reaches 8'hff, it stays 8'hff. Error count value will be reset everytime shallow_lb_en toggle from 0 to 1. Can be used on ATE for BIST.

A.13.1.2 Status

Table 662: POWER_MANAGEMENT_DEVICE_REG Register
Offset: 0x000500C8

Bit	Field	Type/InitVal	Description
Power management register. if power_gating is register controlled, this register is accessable even the power is gated. Refer to device controller integration spec for detail.			
31:16	reserved	RO 0x0	reserved
15	dwack	RO 0x1	dwack DPMU indicates power down or wakeup completion. Default is 1.
14:6	reserved1	RO 0x0	reserved1
5:3	dpreq	RO 0x7	dpreq 1: DPMU requests power on from MPMU. 0: DPMU requests power off from MPMU. Default is 1 for each bit.
2:1	dcreq	RO 0x3	dcreq When set 1, indicate DPMU requests clock recovery from MPMU. When set 0, indicate DPMU requests clock removal from MPMU. Default is 1.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 662: POWER_MANAGEMENT_DEVICE_REG Register (Continued)
Offset: 0x000500C8

Bit	Field	Type/InitVal	Description
0	dwreq	RO 0x1	dwreq During P3 power down or USB2 suspend mode, this signal is de-asserted by USB3 subsystem to indicate the device is in P3 or USb2 suspend mode. By setting this signal to 1, the device indicates to SoC it has received remote wakeup request from USB3 host. This signal also acknowledges the syswreq when SoC requests remote wakeup or disabling USB3 device. Default is 1.

Table 663: POWER_MANAGEMENT_SOC_REG Register
Offset: 0x000500CC

Bit	Field	Type/InitVal	Description
Power management register. if power_gating is register controlled, this register is accessable even the power is gated. Refer to device integration spec for detail.			
31:16	reserved	RO 0x0	reserved
15:13	syspack	RW 0x7	syspack During P3 power down or USB2 suspend mode, this signal is de-asserted to acknowledge the power down.
12:11	syscack	RW 0x3	syscack During P3 power down or USB2 suspend mode, this signal is de-asserted to acknowledge the axi_clk off.
10:6	reserved1	RO 0x0	reserved1 1: MPMU acknowledges clock on request. 0: MPMU acknowledges clock off request. Default is 1.
5:3	syspreq	RW 0x7	syspreq 1: MPMU acknowledges power on request. 0: MPMU acknowledges power off request. Default is 1 for each bit.
2:1	syscreq	RW 0x3	syscreq 1: MPMU completes clock on request. 0: MPMU completes clock off request. Default is 1.

Copyright © 2015 Marvell

Table 663: POWER_MANAGEMENT_SOC_REG Register (Continued)
Offset: 0x000500CC

Bit	Field	Type/InitVal	Description
0	syswreq	RW 0x1	syswreq During P3 power down or USB2 suspend mode, this signal is de-asserted to acknowledge the request of dwreq to power down. By setting this signal to 1, the SoC indicates remote wakeup request to USB3 subsystem. This signal can also be used by SoC to disable USB3 subsystem when SoC de-asserts this signal to 0. Default is 1.

Table 664: LOW_POWER_STATUS_REG Register
Offset: 0x000500D0

Bit	Field	Type/InitVal	Description
if power_gating is register controlled, this register is accessable even the power is gated. This is debug register			
31:7	reserved	RO 0x0	reserved
6	powerpresent	RO 0x1	powerpresent When set, it indicates that the controller detects vbus on.
5:1	present_state	RO 0x0	present_state State machine information for low power DPMU.
0	usb2_run	RO 0x0	usb2_run When set to 1, it indicates the SW has set the usb2_run bit to 1. Used for debug.

Table 665: SOFTWARE_RESET_REG Register
Offset: 0x000500D4

Bit	Field	Type/InitVal	Description
if power_gating is register controlled, this register is accessable even the power is gated.			
31:2	reserved	RO 0x0	reserved
1	core_clk_as_sys_clk_mode	RW 0x0	core_clk_as_sys_clk_mode If IP is configured as supporting "CORE_CLK_AS_SYS_CLK", this bit is then default to 1'b1, which makes IP's SuperSpeed DMA clocked by ss_core_clk. Setting this bit to 0, will result SuperSpeed DMA clocked by ss_sys_clk. Most of customers does not request above configuration. If IP is not configured as "CORE_CLK_AS_SYS_CLK", this bit default is 0. Also toggling this bit results no use.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 665: SOFTWARE_RESET_REG Register (Continued)
Offset: 0x000500D4

Bit	Field	Type/InitVal	Description
0	software_reset	RW 0x0	software_reset this bit is for software to do software reset. Set to 1 to reset all logics in the controller. It is seflf clear.

Table 666: TOP_INTERRUPT_STATUS_REG Register
Offset: 0x000500D8

Bit	Field	Type/InitVal	Description
if power_gating is register controlled, this register is accessable even the power is gated.			
31:6	reserved	RO 0x0	reserved
5	vbus_int	WO 0x0	vbus_int Active high. vbus on or vbus off will trigger this interrupt
4	pme	WO 0x0	pme_int Active high. power management event like dwreq dwack will trigger this interrupt
3	ref_int	RO 0x0	ref_int Active high. Reference clock domain event will trigger this interrupt. this bit is read only, please clear related status register to clear this bit.
2	ss_core_int	RO 0x0	ss_core_int Active high. Core clock domain event will trigger this interrupt. this bit is read only, please clear related status register to clear this bit.
1	ss_sys_int	RO 0x0	ss_sys_int ss_sys clock domain event will trigger this interrupt. this bit is read only, please clear related status register.
0	usb2_int	RO 0x0	usb2_int Active high. USB2 event will trigger this interrupt. this bit is read only, please clear related status register.

Copyright © 2015 Marvell

Table 667: TOP_INTERRUPT_ENABLE_REG Register
Offset: 0x000500DC

Bit	Field	Type/InitVal	Description
if power_gating is register controlled, this register is accessable even the power is gated. This interrupt enable register is reset by only power-on reset and software reset.			
31:6	reserved	RO 0x0	reserved
5	vbus_int_en	RO 0x1	vbus_int_en Interrupt enable for vbus_int event. Set to 1 to enable.
4	pme_en	RW 0x0	pme_en enable bit for pme_int event. Set to 1 to enable.
3	ref_int_en	RW 0x0	ref_int_en enable bit for ref_int event. Set to 1 to enable.
2	ss_core_int_en	RW 0x0	ss_core_int_en enable bit for ss_core_int. Set to 1 to enable.
1	ss_axi_int_en	RW 0x0	ss_axi_int_en enable bit for ss_axi_int. Set to 1 to enable.
0	usb2_int_en	RW 0x0	usb2_int_en enable bit for usb2_int. Set to 1 to enable.

Table 668: PME_INTERRUPT_STATUS_EN_REG Register
Offset: 0x000500E0

Bit	Field	Type/InitVal	Description
if power_gating is register controlled, this register is accessable even the power is gated.			
31:16	reserved	RO 0x0	reserved
15	dwack_int_en	RW 0x0	dwack_int_en for dpmu, when this bit is set, if dwack changes, pme_int will be set to 1
14:4	reserved1	RO 0x0	reserved1
3	dpreq_int_en	RW 0x0	dpreq_int_en for dpmu, when this bit is set, if dpreq changes, pme_int will be set to 1
2:1	reserved2	RO 0x0	reserved2

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 668: PME_INTERRUPT_STATUS_EN_REG Register (Continued)
Offset: 0x000500E0

Bit	Field	Type/InitVal	Description
0	dwreq_int_en	RW 0x0	dwreq_int_en for dpmu, when this bit is set, if dwreq changes, pme_int will be set to 1

Table 669: AHB_TIMEOUT_REG Register
Offset: 0x000500E4

Bit	Field	Type/InitVal	Description
31:16	reserved	RO 0x0	reserved The controller will issue error response if the read data is not ready when this timeout value is reached.
15:0	ahb_timeout_value	RW 0xFF	ahb_timeout_value This is for error handling.

A.13.1.3 PA

Table 670: DSP_DISCONNECT Register
Offset: 0x00058400

Bit	Field	Type/InitVal	Description
This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0 This Register is used by a downstream port following a LINE-RESET to signal a DSP Disconnect. This bit is required only for an USP only. A DSP or a USP in the MPHY.TEST state shall ignore writes to this register. bit[0]: software read only on USP port. If this bit is 1’b1, it means DSP is signaling a DSP disconnect to this USP. Writing 1’b0 shall have no effect bit[7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved
0	DSP_DISCONNECT	RO 0x0	DSP_DISCONNECT SSIC host set the DSP_DISCONNECT bit in RRAP to inidicate DSP disconnection.

Table 671: CONFIGURE_FOR_HS Register
Offset: 0x00058401

Bit	Field	Type/InitVal	Description
This Register is used to direct the Target to update its shadow bank for HS-BURST with the settings that correspond to the SSIC profile supported. The Target shall provide a Write Response to this Command only after the shadow bank registers are updated. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0 bit[0]:Writing 1'b1 directs the Target to update its shadow bank with the settings that correspond to its SSIC profile. Writing 1'b0 shall have no effect bit[7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved
0	CONFIGURE_FOR_HS	RO 0x0	CONFIGURE_FOR_HS SSIC host set the CONFIGURE_FOR_HS bit in RRAP to direct the Target to update the shadow bank for HS-BURST with the settings that correspond to the SSIC profile supported.

Table 672: BURST_CLOSURE Register
Offset: 0x00058402

Bit	Field	Type/InitVal	Description
This Register is used to direct the Target to terminate the PWM-BURST. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0 bit[0]: SSIC host sends Burst_Closure RRAP to set this bit to 1'b1 and device will terminate the PWM-BURST and initiate an RCT to exit LS-MODE. Writing 1'b0 shall have no effect. bit[7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 672: BURST_CLOSURE Register (Continued)
Offset: 0x00058402

Bit	Field	Type/InitVal	Description
0	CONFIGURE_FOR_HS	RO 0x0	CONFIGURE_FOR_HS SSIC host set the BURST_CLOSURE bit in RRAP to terminates the PWM-BURST and initiates an RCT to exit LS-MODE.

Table 673: DISABLE_SCRAMBLING Register
Offset: 0x00058403

Bit	Field	Type/InitVal	Description
This register is used by a DSP to indicate to an USP that data transmission in HS-MODE shall have scrambling disabled. A DSP or a USP in the MPHY.TEST state shall ignore writes to this register. This register is only appliable when controller is working in SSIC mode. Not appliable for SuperSpeed mode. Default value: 0 bit[0]:Writing 1b1 configures the USP to disable HS-MODE scrambling. Writing 1b0 shall have no effect. bit[7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved
0	DISABLE_SCRAMBLING	RO 0x0	DISABLE_SCRAMBLING SSIC host set the DISABLE_SCRAMBLING bit in RRAP to indicate to an USP that data transmission in HS-MODE shall have scrambling disabled. DSP SW should set its local PA DISABLE_SCRAMBLING to 1, if it sets this bit in remote PA to 1.

Copyright © 2015 Marvell

Table 674: DISABLE_STALL_IN_U0 Register
Offset: 0x00058404

Bit	Field	Type/InitVal	Description
This Register is used by a DSP to disable STALL entry in U0 in an USP. A DSP or a USP in the MPHY.TEST state shall ignore writes to this register. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0 bit[0]:Writing 1'b1 to configures the USP to disable STALL entry while in U0. Writing 1'b0 shall have no effect. bit[7:1]: Reserved.			
7:1	Reserved	RO 0x0	Reserved
0	DISABLE_STALL_IN_U0	RO 0x0	DISABLE_STALL_IN_U0 SSIC host set the DISABLE_STALL_IN_U0 bit in RRAP to disable STALL entry in U0 in an USP.

Table 675: DISABLE_LUP_LDN Register
Offset: 0x00058405

Bit	Field	Type/InitVal	Description
This optional Register is used by a DSP to disable the use of the LDN and LUP Link Commands defined in [USB 3.0] to allow additional power optimizations in the M-TX. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0 bit[0]:Writing 1'b1 configures the USP to disable LUP/LDN commands. In addition, the USP shall not flag a tU0LTimeout Error or transition to Recovery upon detecting a missing LUP or LDN. A DSP that generates a RRAP command to set this bit shall not transmit any LUP and LDN link commands. In addition, the DSP shall not flag a tU0LTimeout Error or transition to Recovery upon detecting a missing LUP or LDN. Writing 1'b0 shall have no effect. bit[7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 675: DISABLE_LUP_LDN Register (Continued)
Offset: 0x00058405

Bit	Field	Type/InitVal	Description
0	DISABLE_LUP_LDN	RO 0x0	DISABLE_LUP_LDN SSIC host set the DISABLE_LUP_LDN bit in RRAP to configure the USP to disable LUP/LDN commands. In addition, the USP shall not flag a tU0LTimeout Error . or LDN.

Table 676: TEST_MODE Register
Offset: 0x000584FF

Bit	Field	Type/InitVal	Description
This Register is used by Test Equipment to configure a Target in the MPHY.TEST state. The Target shall complete its configuration to the MPHY.TEST state prior to issuing a Write Response to the command. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0 bit[0]:Writing 1'b1 configures the RRAP Target in the MPHY.TEST state. Writing 1'b0 shall have no effect. bit[7:1]: Reserved.			
7:1	Reserved	RO 0x0	Reserved
0	TEST_MODE	RO 0x0	TEST_MODE Test Equipment set the TEST_MODE bit in RRAP to configure a Target in the MPHY.TEST state.

Table 677: PAIR_CAPABILITY Register
Offset: 0x00058500

Bit	Field	Type/InitVal	Description
This Register is used by a SSIC host to determine the number PAIRs supported by a SSIC device. The DSP only read this register through RRAP. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: HW configured. Bit [0]:A read of this field shall always return a value of 1'b1 Bit [1]:A read of this field shall return a value of 1'b1 if PAIR0 and PAIR1 are supported. Bit [2]:A read of this field shall return a value of 1'b1 if PAIR0, PAIR1, PAIR2 and PAIR3 are supported. bit[7:3]: Reserved.			
7:3	Reserved	RO 0x0	Reserved
2:0	PAIR_CAPABILITY	RO 0x1	PAIR_CAPABILITY SSIC Host read the PAIR_CAPABILITY bit in RRAP to determine the number PAIRs supported by a Target.

Table 678: LOOPBACK_EN Register
Offset: 0x00058E00

Bit	Field	Type/InitVal	Description
This Register is used to enable Loopback Testing for the PAIR under Test. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [0]:Writing 1'b1 enables the loopback mode or the PAIR under Test. Writing 1'b0 shall have no effect. bit[7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 678: LOOPBACK_EN Register (Continued)
Offset: 0x00058E00

Bit	Field	Type/InitVal	Description
0	LOOPBACK_EN	RO 0x0	LOOPBACK_EN Test Equipment set the LOOPBACK_EN bit in RRAP to enable Loopback Testing for the PAIR under Test.

Table 679: RX_BURST_COUNT_0 Register
Offset: 0x00058E01

Bit	Field	Type/InitVal	Description
These Registers RX_BURST_COUNT_0,RX_BURST_COUNT_1,RX_BURST_COUNT_2,RX_BURST_COUNT_3 count the number of bursts by incrementing on each HS-BURST to STALL transition. RX_BURST_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The LSB value of the number of HS-BURST to STALL transitions.			
7:0	RX_BURST_COUNT_0	RO 0x0	RX_BURST_COUNT_0 This register count the number of HS-BURST to STALL transitions.

Copyright © 2015 Marvell

Table 680: RX_BURST_COUNT_1 Register
Offset: 0x00058E02

Bit	Field	Type/InitVal	Description
These Registers RX_BURST_COUNT_0,RX_BURST_COUNT_1,RX_BURST_COUNT_2,RX_BURST_COUNT_3 count the number of bursts by incrementing on each HS-BURST to STALL transition. RX_BURST_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of HS-BURST to STALL transitions.			
7:0	RX_BURST_COUNT_1	RO 0x0	RX_BURST_COUNT_1 This register count the number of HS-BURST to STALL transitions. This register shall be reset only when RX_COUNT_RESET[0] is set.

Table 681: RX_BURST_COUNT_2 Register
Offset: 0x00058E03

Bit	Field	Type/InitVal	Description
These Registers RX_BURST_COUNT_0,RX_BURST_COUNT_1,RX_BURST_COUNT_2,RX_BURST_COUNT_3 count the number of bursts by incrementing on each HS-BURST to STALL transition. RX_BURST_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of HS-BURST to STALL transitions.			
7:0	RX_BURST_COUNT_2	RO 0x0	RX_BURST_COUNT_2 This register count the number of HS-BURST to STALL transitions. This register shall be reset only when RX_COUNT_RESET[0] is set.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 682: RX_BURST_COUNT_3 Register
Offset: 0x00058E04

Bit	Field	Type/InitVal	Description
These Registers RX_BURST_COUNT_0,RX_BURST_COUNT_1,RX_BURST_COUNT_2,RX_BURST_COUNT_3 count the number of bursts by incrementing on each HS-BURST to STALL transition. RX_BURST_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of HS-BURST to STALL transitions.			
31:8	Reserved	RSVD 0x0	Reserved
7:0	RX_BURST_COUNT_3	RO 0x0	RX_BURST_COUNT_3 This register count the number of HS-BURST to STALL transitions. This register shall be reset only when RX_COUNT_RESET[0] is set.

Table 683: RX_ERR_COUNT_0 Register
Offset: 0x00058E05

Bit	Field	Type/InitVal	Description
These Registers RX_ERR_COUNT_0,RX_ERR_COUNT_1,RX_ERR_COUNT_2,RX_ERR_COUNT_3 the number of errors by incrementing on each HS-BURST to STALL transition for which an MK0 was not detected. RX_ERR_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of errors HS-BURST to STALL transitions for which an MK0 was not detected.			
7:0	RX_ERR_COUNT_0	RO 0x0	RX_ERR_COUNT_0 This register count the number of HS-BURST to STALL transitions for which the MK0 is not detected. This register shall be reset only when RX_COUNT_RESET[0] is set.

Copyright © 2015 Marvell

Table 684: RX_ERR_COUNT_1 Register
Offset: 0x00058E06

Bit	Field	Type/InitVal	Description
These Registers RX_ERR_COUNT_0,RX_ERR_COUNT_1,RX_ERR_COUNT_2,RX_ERR_COUNT_3 the number of errors by incrementing on each HS-BURST to STALL transition for which an MK0 was not detected. RX_ERR_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of errors HS-BURST to STALL transitions for which an MK0 was not detected.			
7:0	RX_ERR_COUNT_1	RO 0x0	RX_ERR_COUNT_1 This register count the number of errors HS-BURST to STALL transitions for which the MK0 is not detected. This register shall be reset only when RX_COUNT_RESET[0] is set.

Table 685: RX_ERR_COUNT_2 Register
Offset: 0x00058E07

Bit	Field	Type/InitVal	Description
These Registers RX_ERR_COUNT_0,RX_ERR_COUNT_1,RX_ERR_COUNT_2,RX_ERR_COUNT_3 the number of errors by incrementing on each HS-BURST to STALL transition for which an MK0 was not detected. RX_ERR_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of errors HS-BURST to STALL transitions for which an MK0 was not detected.			
7:0	RX_ERR_COUNT_2	RO 0x0	RX_ERR_COUNT_2 This register count the number of errors HS-BURST to STALL transitions for which the MK0 is not detected. This register shall be reset only when RX_COUNT_RESET[0] is set.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 686: RX_ERR_COUNT_3 Register
Offset: 0x00058E08

Bit	Field	Type/InitVal	Description
These Registers RX_ERR_COUNT_0,RX_ERR_COUNT_1,RX_ERR_COUNT_2,RX_ERR_COUNT_3 the number of errors by incrementing on each HS-BURST to STALL transition for which an MK0 was not detected. RX_ERR_COUNT_0 is the LSB. This register shall be reset only when RX_COUNT_RESET[0] is set. This register is not affected by a LINE-RESET issued by Test Equipment. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [7:0]:The value of the number of errors HS-BURST to STALL transitions for which an MK0 was not detected.			
31:8	Reserved	RSVD 0x0	Reserved
7:0	RX_ERR_COUNT_3	RO 0x0	RX_ERR_COUNT_3

Table 687: RX_COUNT_RESET Register
Offset: 0x00058E09

Bit	Field	Type/InitVal	Description
This register is used to reset the RX_BURST_COUNT and the RX_ERR_COUNT registers. This register is write only and self-clearing. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [0]:Writing 1’b1 resets the RX_BURST_COUNT and the RX_ERR_COUNT registers. Bit [7:1]: Reserved.			
7:1	Reserved	RO 0x0	Reserved
0	RX_COUNT_RESET	WO 0x0	RX_COUNT_RESET The software set the RX_COUNT_RESET bit to clear the registers RX_BURST_COUNT_x and RX_ERR_COUNT_x. The register is self-clearing.

Copyright © 2015 Marvell

Table 688: TX_COMP_MODE_EN Register
Offset: 0x00058E0A

Bit	Field	Type/InitVal	Description
This register is used to enable M-TX of the PAIR under Test in TX Compliance mode. Writes to this register are undefined unless the DUT is in the MPHY.TEST state. This register is only appliable when controller is working in SSIC mode. Not appliable for Superspeed mode. Default value: 0. Bit [0]:Writing 1'b1 enables Tx Compliance Mode for the PAIR under Test. Writing 1'b0 shall be ignored and Reads shall return zero values. Bit [7:1]: Reserved.			
31:8	Reserved	RSVD 0x0	Reserved
7:1	Reserved	RO 0x0	Reserved
0	TX_COMP_MODE_EN	RO 0x0	TX_COMP_MODE_EN The Test Equipment set the TX_COMP_MODE _EN bit in RRAP to enable M-TX of the PAIR under Test in TX Compliance mode.

Table 689: ANALOG_LOOPBACK_EN Register
Offset: 0x00058E0B

Bit	Field	Type/InitVal	Description
This register is used to place the M-RX of the PAIR under Test in Analog Loopback mode. Writes to this register are undefined unless the DUT is in the MPHY.TEST state. This register is only applicable when controller is working in SSIC mode. Not appliable for SuperSpeed mode. Default value: 0. Bit [0]: Writing 1b1 enables Analog Loopback Mode for the PAIR under Test. Writing 1b0 will be ignored and Reads will return zero values. Bit [7:1]: Reserved.			
7:1	Reserved	RO 0x0	Reserved
0	ANALOG_LOOPBACK_EN	RO 0x0	ANALOG_LOOPBACK_EN The Test Equipment set the ANOLOG_LOOPBACK _EN bit in RRAP to place the M-RX of the PAIR under Test in Analog Loopback mode.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 690: PM_CLOCK_GATE_EN Register
Offset: 0x00058F00

Bit	Field	Type/InitVal	Description
This register is used to for SW to enable/disable clock gating in PA module, and MPHY clock shut down during low power mode. Note: Current Marvell MPHY may not support clock shut down in Hibern8.			
31:8	Reserved	RSVD 0x0	Reserved
7:4	Reserved	RO 0x0	Reserved
3	PA_U3_CLK_DISABLE_EN	RW 0x0	PA_U3_CLK_DISABLE_EN bit3: For MPY TX_Symobl clock disabling in U3 state. Set to 1 to disable clock from MPHY. In this case, MPHY PLL is disabled. PIPE_CLK to controller is not available.
2	PA_U2_CLK_DISABLE_EN	RW 0x0	PA_U2_CLK_DISABLE_EN bit2: For MPHY output TX_Symbol clock disabling in U2. Set to 1 to disable clock from MPHY. In this case, MPHY PLL is disabled. PIPE_CLK to controller is not available.
1	PA_U2_CLK_GATE_EN	RW 0x0	PA_U2_CLK_GATE_EN bit1: For PA clock gating in U2 state. Set to 1 to gate mcore_clk, mtx_clk, mpipe_clk inside PA when device is in U2. PIPE_CLK to controller should be stillavailable.
0	PA_U1_CLK_GATE_EN	RW 0x0	PA_U1_CLK_GATE_EN For PA clock gating in U1 state. Set to 1 to gate mcore_clk, mtx_clk, mpipe_clk inside PA when device is in U1. PIPE_CLK to controller should be still available.

Table 691: Timer_Hibern8_Enter_TX Register
Offset: 0x00058F02

Bit	Field	Type/InitVal	Description
Default: d1 - 1us based on 1us clock pulse generated by PA. Bit[7:0]: Timer value The timer of Hibern8_Enter_TX is in the range of 50-1000ns defined by [M-PHY] specification. The timer should be run by 1us clock pulse. The value needs to be adjusted based on M-PHY parameter.			
31:8	Reserved	RSVD 0x0	Reserved
7:0	Timer_Hibern8_Enter_TX	RW 0x1	Timer_Hibern8_Enter_TX This register is the timer of Hibern8_Enter_TX, based on 1us clock enable pulse.

Copyright © 2015 Marvell

Table 692: Timer_Hibern8_Residency Register
Offset: 0x00058F04

Bit	Field	Type/InitVal	Description
This register is the timer of Hibern8_Residency . It is counted for 100us by SSIC profile definition. It is adjustable by SW for flexibility. The timer should be run based on the 1us timer. The value needs to be adjusted based on M-PHY parameter and ref_clk frequency. Default value: d100- 1us based on the unit 1us timer.			
7:0	Timer_Hibern8_Residency	RW 0x64	Timer_Hibern8_Residency This register is the timer of Hibern8_Residency, based on the 1us timer.

Table 693: Num_of_Cycles_for_1us Register
Offset: 0x00058F06

Bit	Field	Type/InitVal	Description
This register is the timer of 1us based on the reference clk. It is counted for clock cycles in 1us. It is adjustable by SW for flexibility. Default value: d26- 1us based on the ref_clk 26MHz.			
31:8	Reserved	RSVD 0x0	Reserved
7:0	Num_of_Cycles_for_1us	RW 0x1A	Num_of_Cycles_for_1us This register is the timer of 1us, based on the ref_clk.

Table 694: MPHY_RMMI_Register_Base_Addr
Offset: 0x00058F08

Bit	Field	Type/InitVal	Description
This register is the base address for MPHY registers defined by MIPI standard. When PA accesses those registers in MPHY via RMMI control interface, it should add the base address to the address defined by MIPI.			
31:16	Reserved	RSVD 0x0	Reserved
15:0	MPHY_RMMI_Register_Base_Addr	RW 0x100	MPHY_RMMI_Register_Base_Addr This register is the base address for the APB to access the Marvell defined registers in MPHY through RMMI. When doing this should add this base address.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 695: Num_of_Cycles_for_10us Register
Offset: 0x00058F0A

Bit	Field	Type/InitVal	Description
This register is the timer of 10us based on the reference clk. It is counted for clock cycles in 10us. It is adjustable by SW for flexibility. Default value: d260- 1us based on the ref_clk 26MHz.			
15:0	Num_of_Cycles_for_10us	RW 0x104	Num_of_Cycles_for_10us This register is the timer of 10us, based on the ref_clk.

Table 696: SSIC_Dev_Timer_Tretrain Register
Offset: 0x00058F0E

Bit	Field	Type/InitVal	Description
This register is the timer of the tRetrain timer.It is adjustable by SW for flexibility. Default value: d48- based on the unit 1us.			
7:0	Ssic_Dev_Timer_Tretrain	RW 0x30	Ssic_Dev_Timer_Tretrain This register is the timer of 48us, based on the unit 1us.

Table 697: SSIC_Dev_Pollingactive_Residency Register
Offset: 0x00058F10

Bit	Field	Type/InitVal	Description
This register is the timer of the PollingActive timeout timer. It is adjustable by SW for flexibility. Default value: d58- based on the unit 1ms.			
31:8	Reserved	RSVD 0x0	Reserved
7:0	Ssic_Dev_Pollingactive_Residency	RW 0x3A	Ssic_dev_Pollingactive_Residency This register is the timer of 58ms, based on the unit 1ms.

Copyright © 2015 Marvell

Table 698: SSIC_Control_Status_Register
Offset: 0x00058F20

Bit	Field	Type/InitVal	Description
This register record the status of the HW.The register can be read by SW and is cleared by write 1 to according bit. bit[15:9]: Reserved. bit[8]: USP_Disconnect_Done This bit should only be written by HW.PA sets this bit to 1 to indicate the completion of USP disconnect.SW write 1 to clear. bit[7:3]: Reserved. bit[2]: Elastic buffer underflow The error is received from PHY when operating in HS-MODE. SW write 1 to clear. bit[1]: Elastic buffer overflow The error is received from PHY when operating in HS-MODE. SW write 1 to clear. bit[0]: 8B/10B decode error The error is received from PHY when operating in HS-MODE. SW write 1 to clear. Default value: 0			
31:16	Reserved	RSVD 0x0	Reserved
15:9	Reserved_2	RO 0x0	Reserved_2
8	USP_Disconnect_Done	WO 0x0	USP_Disconnect_Done This register record the usp_disconnect_done bit8. SW write 1 to clear the according bit.
7:3	Reserved_1	RO 0x0	Reserved_1
2	Elastic_buffer_underflow	WO 0x0	Elastic buffer underflow This register record the underflow errors bit2. SW write 1 to clear the according bit.
1	Elastic_buffer_overflow	WO 0x0	Elastic buffer overflow This register record the elastic buffer overflow bit1. SW write 1 to clear the according bit.
0	eightB_tenB_decode_error	WO 0x0	8B_10B_decode_error This register record the 8/10 decode error bit0. SW write 1 to clear the according bit.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 699: SSIC_Status_Interrupt_Enable_Register
Offset: 0x00058F22

Bit	Field	Type/InitVal	Description
Default: 0 SW sets the respective bit to 1 to enable interrupt for the events listed in ssic_control_status register.			
31:16	Reserved	RSVD 0x0	Reserved
15:9	Reserved_2	RO 0x0	Reserved_2
8	USP_Disconnect_Done_interrupt_enable	RW 0x0	USP_Disconnect_Done_interrupt_enable This register record the usp_disconnect_done bit8. SW write 1 to clear the according bit.
7:3	Reserved_1	RO 0x0	Reserved_1
2	Elastic_buffer_underflow_interrupt_eanble	RW 0x0	Elastic buffer underflow_interrupt_enable This register record the underflow errors bit2. SW write 1 to clear the according bit.
1	Elastic_buffer_overflow_interrupt_enable	RW 0x0	Elastic buffer overflow_interrupt_enable This register record the elastic buffer overflow bit1. SW write 1 to clear the according bit.
0	eightB_tenB_decode_error_interrupt_enable	RW 0x0	8B10B decode error_interrupt_enable This register record the 8/10 decode error bit0. SW write 1 to clear the according bit.

Copyright © 2015 Marvell

Table 700: Timer_Tx_Tactive_Time Register
Offset: 0x00058F28

Bit	Field	Type/InitVal	Description
Bit[15:0]: Timer value Tactive Time is used by MPHY to determine how long to drive DIF-N before driving DIF-P for Line Reset. It is also used to control the length to drive DIF-N for Hibern8 exit. If TX_Controlled_ActTimer signal on RMMI interface is tied to ?1?, the value for this register can be set to a small value (or keep it as default) since Tactive Time is controlled by MPHY. If TX_Controlled_ActTimer signal on RMMI interface is tied to ?0?, the value for this register should be set based on MPHY TX configuration attributes (Tx_Advanced_Granularity_Setting, Tx_Advanced_Granularity, and Tx_Min_ActiveTime). In this case, the Tactive Time is controlled by PA. Since Marvell?s MPHY supports controlling Tactive_time, TX_Controlled_ActTimer signal on the RMMI interface should be tied to ?1? as a default configuration. The timer should be run by ref_clk. The value needs to be adjusted if ref_clk is not 26Mhz.			
31:16	Reserved	RSVD 0x0	Reserved
15:0	Timer_Tx_Tactive_Time	RW 0x1	Timer_Tx_Tactive_Time This register is based on the unit 1us.

Table 701: Timer_Rx_Tactive_Time Register
Offset: 0x00058F2A

Bit	Field	Type/InitVal	Description
Default: d100 (100us). The value is based on 1us timer. Bit[15:0]: Timer value. This value should be set based on MPHY RX configuration attributes (Rx_Advanced_Granularity_Capability, Rx_Advanced_Min_ActiveTime_capability).			
31:16	Reserved	RSVD 0x0	Reserved
15:0	Timer_Rx_Tactive_Time	RW 0x64	Timer_Rx_Tactive_Time This register is based on the unit 1us.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

A.13.1.4 Reference

Table 702: DEV_INFO_REG Register
Offset: 0x00050000

Bit	Field	Type/InitVal	Description
31	ssic_mode_en	RW 0x0	ssic_mode_en ssic_mode_en is only appliable to device controller which supports both Superspeed and SSIC mode. For Superspeed only device controller, this bit is reserved. ssic_mode_en is used by software to select iif controller is working in Superspeed mode or SSIC mode. Software shall issue software reset before mode switching. By default, controller is working in Superspeed mode. Software shall set ssic_mode_en to 1 before asserting soft_connect to enable connection to host. 1’b0 : Controller is working in Superspeed mode. This is the default value. 1’b1: Controller is working in SSIC mode. 1’b1: Controller is working in SSIC mode.
30:18	Reserved_1	RO 0x0	Reserved_1
17:16	device_speed	RW 0x0	device_speed Software can program this field to make the core operate in a specific mode. The encoding is as follows: Bits [17:16] 00: USB3 device - Device supports three types of speeds- SuperSpeed, High speed and full speed 01: Reserved. 10: USB2 device only - Supports only high and full speed. 11: disable usb2 and superspeed.
15:0	version_id	RO 0xD2	version_id The Vision ID is the release version

Table 703: DEV_TAG_REG Register
Offset: 0x00050004

Bit	Field	Type/InitVal	Description
31:0	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 704: LOW_POWER_LFPS_SIGNAL_VAL_REG_1 Register
Offset: 0x00050008

Bit	Field	Type/InitVal	Description
Internal use. Please refer to USB3 specification, 6.9.2 for the usage of the parameters.			
31:28	Reserved_1	RO 0x0	Reserved_1
27:24	lfps_tx_u1_t12_t11	RW 0x6	lfps_tx_u1_t12_t11 Transmitter side of LFPS t12-t11 duration for U1 exit (1unit = 100ns)
23:16	lfps_rx_u3_t13_t11	RW 0x2	lfps_rx_u3_t13_t11 Receiver side of LFPS t13-t11 duration for U3 exit (1unit = 100us)
15:13	Reserved_2	RO 0x0	Reserved_2
12:8	lfps_rx_u2_t13_t11	RW 0x6	lfps_rx_u2_t13_t11 Receiver side of LFPS t13-t11 duration for U2 exit (1unit = 100us)
7:0	Reserved_3	RO 0x0	Reserved_3

Table 705: LOW_POWER_LFPS_SIGNAL_VAL_REG_2 Register
Offset: 0x0005000C

Bit	Field	Type/InitVal	Description
Internal use. Please refer to the USB3 Specification 6.9.2 for the usage of the parameters			
31:24	reserved_1	RO 0x0	reserved_1
23:16	lfps_rx_u3_t11_t10	RW 0x3	lfps_rx_u3_t11_t10 Receiver side of LFPS t11-t10 duration for U3 exit (1unit = 100ns)
15:8	lfps_rx_u2_t11_t10	RW 0x3	lfps_rx_u2_t11_t10 Receiver side of LFPS t11-t10 duration for U2 exit (1unit = 100ns)
7:4	reserved_2	RO 0x0	reserved_2
3:0	lfps_rx_u1_t11_t10	RW 0x3	lfps_rx_u1_t11_t10 Value of receiver side of LFPS t13-t11 duration for u1 exit (in 100ns unit), when u2_inactive_timeout is enabled.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 706: LOW_POWER_LFPS_SIGNAL_VAL_REG_3 Register
Offset: 0x00050010

Bit	Field	Type/InitVal	Description
Internal use. Please refer to USB3 Specification 6.9.2 for the usage of the parameters.			
31:15	lfps_tx_u3_t11_t10	RW 0x1869E	lfps_tx_u3_t11_t10 Transmitter side of LFPS t11-t10 duration for U3 exit (1unit = 100ns)
14:0	lfps_tx_u2_t11_t10	RW 0x4E1E	lfps_tx_u2_t11_t10 Transmitter side of LFPS t11-t10 duration for U2 exit (1unit = 100ns)

Table 707: LOW_POWER_LFPS_SIGNAL_VAL_REG_4 Register
Offset: 0x00050014

Bit	Field	Type/InitVal	Description
Internal use. Refer to USB3 Specification 6.9.2 for the usage of the parameters.			
31:15	lfps_tx_u3_t12_t11	RW 0x1869D	lfps_tx_u3_t12_t11 Transmitter side of LFPS t12-t11 duration for U3 exit (1unit = 100ns)
14:0	lfps_tx_u2_t12_t11	RW 0x4E1D	lfps_tx_u2_t12_t11 Transmitter side of LFPS t12-t11 duration for U2 exit (1unit = 100ns)

Table 708: LOW_POWER_LFPS_SIGNAL_VAL_REG_5 Register
Offset: 0x00050018

Bit	Field	Type/InitVal	Description
Internal use. Please rrefer to USB3 Specification 6.9.2 for details			
31:16	lfps_tx_u1_t11_t10	RW 0x4E1E	lfps_tx_u1_t11_t10 Transmitter side of LFPS t11-t10 duration for U1 exit (1unit = 100ns)
15:0	lfps_rx_u1_t13_t11	RO 0x7	lfps_rx_u1_t13_t11 Receiver side of LFPS t13-t11 duration for U1 exit, in 100ns unit. This value comes from lfps_rx_u1_t13_t11_val2 (LOW_POWER_LFPS_SIGNAL_VAL_REG_5 register [31:16]) when u2_inactive_timeout is enabled. This value comes from lfps_rx_u1_t13_t11_val1 (LOW_POWER_LFPS_SIGNAL_VAL_REG_5 register [15:0]) when u2_inactive_timeout is disabled.

Copyright © 2015 Marvell

Table 709: LOW_POWER_LFPS_SIGNAL_VAL_REG_6 Register
Offset: 0x0005001C

Bit	Field	Type/InitVal	Description
Internal use. Please refer to USB3 Specification 6.9.2 for details.			
31:16	lfps_rx_u1_t13_t11_val2	RW 0x1338	lfps_rx_u1_t13_t11_val2 Value of receiver side of LFPS t13-t11 duration for u1 exit (in 100ns unit), when u2_inactive_timeout is enabled.
15:0	lfps_rx_u1_t13_t11_val1	RW 0x7	lfps_rx_u1_t13_t11_val1 Value of receiver side of LFPS t13-t11 duration for u1 exit (in 100ns unit), when u2_inactive_timeout is disabled.

Table 710: COUNTER_PULSE_VALUE_REG Register
Offset: 0x00050020

Bit	Field	Type/InitVal	Description
31:24	usb3_counter_pulse_100ns	RW 0x5	usb3_counter_pulse_100ns Indicates number of ref clock cycles for 2*100ns. Formula: value ~= 2*100ns/(ref_clk period)
23:20	Reserved_1	RO 0x0	Reserved_1
19:16	usb3_counter_pulse_10ms	RW 0x9	usb3_counter_pulse_10ms Time counter. (1unit = 1ms). Internal use
15:12	usb3_counter_pulse_1ms	RW 0x9	usb3_counter_pulse_1ms Time counter for 1ms (1unit = 100us). Internal use.
11:8	usb3_counter_pulse_100us	RW 0x9	usb3_counter_pulse_100us Time counter for 100us (1unit = 10us). Internal use.
7:4	usb3_counter_pulse_10us	RW 0x9	usb3_counter_pulse_10us Time counter for 10us (1unit = 1us)
3:0	usb3_counter_pulse_1us	RW 0x9	usb3_counter_pulse_1us Time counter for 1us (1unit = 100ns). Internal use.

Table 711: REF_INT_STATUS_REG Register
Offset: 0x00050024

Bit	Field	Type/InitVal	Description
31	usb2_cnt_stat	WO 0x0	usb2_cnt_stat A usb2_vbus_on interrupt has occurred. Indicates that SuperSpeed connection failed or SuperSpeed connection was not attemptted yet.
30	usb2_discnt_stat	WO 0x0	usb2_discnt_stat A usb2_vbus_off interrupt has occurred. Indicates that SuperSpeed is connected.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 711: REF_INT_STATUS_REG Register (Continued)
Offset: 0x00050024

Bit	Field	Type/InitVal	Description
29	resume_rec	WO 0x0	resume_rec Indicates that SuperSpeed/SSIC device controller has been resumed (from U3).
28	suspend_rec	WO 0x0	suspend_rec Indicates that SuperSpeed/SSIC device controller has been suspended (to U3).
27	reset_rec	WO 0x0	reset_rec Indicates that the device received a warm reset.
26	power_on_rec	WO 0x0	power_on_rec This bit is used to indicate that the controller detects vbus on.
25	power_off_rec	WO 0x0	power_off_rec this bit is used to indicate that the controller detects vbus off.
24	ssic_disable_rec	WO 0x0	ssic_disable_rec ssic_disabled_rec bit is only appliable for device controller which could support both Superpseed and SSIC working mode. For Superspeed only device controller, this bit is reserved. Indicates device controller has entered SS.Disabled state due to UPS disconnect or Polling handshake timeout. Once SSIC device controller enters SS.Disabled state, host could not bring it out. There are two conditions which will bring the device LTSSM out of SS.Disabled state: 1. SSIC USP re-connect: When SoC asserts sysreq or SW writes sysreq bit, a HW reset will be generated by reset_gen to reset most of logic in device controller except DPMU. This action will bring the LTSSM from SS.Disabled to RX.Detect. 2. SW reset or system reset to device controller. This bit is not appliable in Superspeed mode.For SSIC host mode, it uses status bits in xHCI PORTSC register.
23:0	Reserved_1	RO 0x0	Reserved_1

Table 712: REF_INT_ENABLE_REG Register
Offset: 0x00050028

Bit	Field	Type/InitVal	Description
This interrupt enable register is reset by power-on reset, software_reset, and internal generated hardware reset (caused by SoC disable/enable device, device unplug/plug)			
31	usb2_cnt_stat_en	RW 0x0	usb2_cnt_stat_en usb2_cnt_stat enable signal. Set to 1 to enable.

Table 712: REF_INT_ENABLE_REG Register (Continued)
Offset: 0x00050028

Bit	Field	Type/InitVal	Description
30	usb2_discnt_stat_en	RW 0x0	usb2_discnt_stat_en usb2_discnt_stat enable signal. Set to 1 to enable
29	resume_rec_en	RW 0x0	resume_rec_en resume_rec enable signal. Set to 1 to enable
28	suspend_rec_en	RW 0x0	suspend_rec_en suspend_rec enable signal. Set to 1 to enable.
27	reset_rec_en	RW 0x0	reset_rec_en reset_rec enable signal. Set to 1 to enable
26	power_on_rec_en	RW 0x0	power_on_rec_en power_on_rec enable signal. Set to 1 to enable.
25	power_off_rec_en	RW 0x0	power_off_rec_en power_off_rec enable signal. Set to 1 to enable.
24:0	Reserved_1	RO 0x0	Reserved_1

Table 713: USB3_GLOBAL_CONTROL_STATUS_REG Register
Offset: 0x0005002C

Bit	Field	Type/InitVal	Description
31	soft_connect	RW 0x0	soft_connect Soft connect bit, active high. If this bit is not asserted, device will appear not connected to far end link partner.
30	reserved_3	RO 0x0	reserved_3
29	safe_mode	RW 0x1	safe_mode If this bit is set, DPMU_P3 is bypassed. From DPMU side, device only enters P2, never enters P3.
28	swrst_with_phyrst	RW 0x0	swrst_with_phyrst If this bit is set, device will reset PHY when a software reset occurs.
27	soc_register_access	RW 0x1	soc_register_access if this bit is set, device will use register set, instead of pin set for low power management
26	usb3_ss_clk_gating_en	RW 0x1	usb3_ss_clk_gating_en If this bit is set to 1, the clocks used for SuperSpeed will be gated when the controller is working in USB2 mode
25	usb2_clk_gating_en	RW 0x1	usb2_clk_gating_en If this bit is set to 1, the clocks for USB2 will be gated when the controller is working in SuperSpeed mode.
24:16	Reserved_2	RO 0x0	Reserved_2
15:12	ssfail_cnt	RO 0x0	ssfail_cnt Internal counter that records how many times SuperSpeed failed to connect.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 713: USB3_GLOBAL_CONTROL_STATUS_REG Register (Continued)
Offset: 0x0005002C

Bit	Field	Type/InitVal	Description
11:8	reg_ssfail_cnt	RW 0x3	reg_ssfail_cnt If the value of this field is not 0, the speed switch SuperSpeed disable counter will be enabled. reg_ssfail_cnt is the counter limit for SuperSpeed disable count during speed switch. Once the limit is reached, device will never switch to SuperSpeed upon USB2 bus reset. Unplug or disconnect will reset this count down.
7	usb3_lfps_exit_ux	RW 0x0	usb3_lfps_exit_ux SW sets this bit to 1 to initiate U3 exit. Cleared by HW when device is out of U3.
6:5	reserved_1	RO 0x0	reserved_1
4	usb2_atmpss_fail	RO 0x0	usb2_atmpss_fail Indicating last attempt from USB2 to SuperSpeed switch failed.
3	ss_vbus	RO 0x0	ss_vbus Indicating SuperSpeed mode.
2	powerpresent	RO 0x0	powerpresent Superspeed mode: real vbus SSIC mode: This bit will be set when host initiates connection to device (device detected Hibern8 exit in START state). This bit will be cleared after device controller finished the DSP disconnect or USP disconnect which means host is not connected to device any more.
1	reserved	RO 0x0	reserved
0	usb2_bus_reset	RW 0x0	usb2_bus_reset

Table 714: SYSTEM_DEBUG_REG Register
Offset: 0x00050030

Bit	Field	Type/InitVal	Description
Debug register. Internal use.			
31:29	Reserved_1	RO 0x0	Reserved_1

Table 714: SYSTEM_DEBUG_REG Register (Continued)
Offset: 0x00050030

Bit	Field	Type/InitVal	Description
28:24	ltssm_state	RO 0x0	ltssm_state This field indicates the LTSSM current state as received from LTSSM in the MAC layer. It is a read only register to the application. The encodings are as follows : 'h1 : SS.Disabled 'h2 : SS.Inactive.Disconnect.Detect 'h3 : SS.Inactive.Quiet 'h4 : Rx.Detect.Reset 'h5 : Rx.Detect.Active 'h6 : Rx.Detect.Quite 'h7 : Polling.LFPS 'h8 : Polling.RxEQ 'h9 : Polling.Active 'hA : Polling.Configuration 'hB : Polling.Idle 'hC : U0 'hD : U1 'hE : U2 'hF : U3 'h10 : Compliance 'h11 : Recovery.Active 'h12 : Recovery.Configuration 'h13 : Recovery.Idle 'h14 : Loopback.Active 'h15 : Loopback.Exit 'h16 : HotReset.Active 'h17 : HotReset.Exit Other : Reserved

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 714: SYSTEM_DEBUG_REG Register (Continued)
Offset: 0x00050030

Bit	Field	Type/InitVal	Description
23:8	Reserved_2	RO 0x0	Reserved_2
7:4	dpmu_state	RO 0x0	dpmu_state This field shows usb3_phy_u3_wrapper states, for debug purpose: `h0: IDLE  `h1: P3_ENTRY_REQ `h2: P3_ENTRY_DONE  `h3: WAIT_PHYSTATUS `h4: P3_STATE  `h5: P3_EXIT_REQ `h6: P3_EXIT_WAIT  `h7: P3_EXIT_DONE `h8: P3_EXIT_DONE_1  `h9: WAIT_FOR_SWITCH `ha: WAIT_FOR_SWITCH_1  `hb: WAIT_FOR_SWITCH_2 `hc: WAIT_FOR_SWITCH_3
3:0	speed_switch_state	RO 0x1	speed_switch_state This field shows speed switch FSM state: `h0: POWERED_OFF  `h1: ATP_SS_CNT `h2: CNTD_USB2  `h3: CNTD_SS `h4: CNTD_USB2_ATP_SS  `h5: SS_TRAINING

Table 715: EP0_ACK_TIMEOUT_VALUE_REG Register
Offset: 0x00050034

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1

Copyright © 2015 Marvell

Table 715: EP0_ACK_TIMEOUT_VALUE_REG Register (Continued)
Offset: 0x00050034

Bit	Field	Type/InitVal	Description
15:0	ep0_ack_timeout_value	RW 0xFF	ep0_ack_timeout_value Error Handling: The timeout value for EP0, if device does not receive ACK from host after this value, the transaction will be aborted.

Table 716: ACK_TIMEOUT_VALUE_REG Register
Offset: 0x00050038

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1 Error handling: The timeout value for EPs other than EP0, if device does not receive ACK from host after this value, the transaction will be aborted.
15:0	ack_timeout_value	RW 0xFFFF	ack_timeout_value

Table 717: RX_IDLE_DELAY_COUNTER_REG Register
Offset: 0x0005003C

Bit	Field	Type/InitVal	Description
Internal use			
31:16	Reserved_1	RO 0x0	Reserved_1 Reserved
15:0	rx_idle_delay_counter	RW 0xA	rx_idle_delay_counter this counter is used to avoid short LFPS in U3. HW will count default 10 ref_clk cycles before it delares it is wakeup siganl.

Table 718: COUNTER_PULSE_VALUE_REG_1 Register
Offset: 0x00050040

Bit	Field	Type/InitVal	Description
31:24	usb3_counter_pulse_256us	RW 0xFF	usb3_counter_pulse_256us Time Counter for 256us. 1 unit = 1us. Internal use
23:2	Reserved_1	RO 0x0	Reserved_1
1	no_gating	RW 0x0	no_gating For debug. Internal use.
0	bk_door	RW 0x0	bk_door For debug. Internal use.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 719: TEST_BUS_MUX_REG Register
Offset: 0x00050044

Bit	Field	Type/InitVal	Description
31:8	Reserved_1	RO 0x0	Reserved_1 The mux setects which group of signals goes to test bus pins. Please refer to Marvelll USB3 Device Test Bus Address for signal group selection. M
7:0	test_bus_mux	RW 0x0	test_bus_mux

Table 720: CLOCK_GATING_ENABLE_REG Register
Offset: 0x00050048

Bit	Field	Type/InitVal	Description
This is power management register. Please refer to device integration spec, power management chapter for clock gating requirement.			
31:14	Reserved_1	RO 0x0	Reserved_1
13	sleep_clk_en	RW 0x0	sleep_clk_en If this bit is set to 1, the controller performs handshake with SoC to switch ref_clk to sleep_clk during power down mode. Refer
12	l2_usb2_dma_clk_gating_en	RW 0x0	l2_usb2_dma_clk_gating_en When set to 1, USB2_dma_clk and internal axi_clk are gated in L2 state. If power gating is enabled in D3, these clock need to be gated regardless this bit is set or not. this bit is set or not.
11	l1_usb2_dma_clk_gating_en	RW 0x0	l1_usb2_dma_clk_gating_en This field is reserved.
10	Reserved_2	RO 0x0	Reserved_2
9	phy_p3_ref_clk_gating_en	RW 0x0	phy_p3_ref_clk_gating_en When set to 1, the controller performs handshaking with SS PHY for SS PHY to switch to use sleep_clk instead of ref_clock in SS PHY in P3 state.
8	p3_ref_clk_gating_en	RW 0x0	p3_ref_clk_gating_en When set to 1, ref clock to USB3 SS MAC is gated in P3 state. If power gating is enabled in D3, this clock needs to be gated regardless the bit is set or not.

Copyright © 2015 Marvell

Table 720: CLOCK_GATING_ENABLE_REG Register (Continued)
Offset: 0x00050048

Bit	Field	Type/InitVal	Description
7	d3_axi_clk_tree_gating_en	RW 0x0	d3_axi_clk_tree_gating_en When set to 1, DPMU should request SoC to turn of axi_clk from SoC side in P3 so that the whole sys_clk clock tree can be disabled in power down mode.
6	p3_ss_dma_clk_gating_en	RW 0x0	p3_ss_dma_clk_gating_en When set to 1, ss_dma_clk and internal axi_clk are gated in P3 state. If power gating is enabled in D3, the clocks need to be gated regardless this bit is set or not.
5	d2_axi_clk_tree_gating_en	RW 0x0	d2_axi_clk_tree_gating_en When set to 1, device controller requests SoC for external AXI clock gating in U2.
4	d1_axi_clk_tree_gating_en	RW 0x0	d1_axi_clk_tree_gating_en When set to 1, device request external AXI clock gating in U1
3	p2_clk_gating_en	RW 0x0	p2_clk_gating_en When set to 1, clocks for USB3 SS core and DMA are gated in P2 state.
2:1	Reserved_4	RO 0x0	Reserved_4
0	p1_clk_gating_en	RW 0x0	p1_clk_gating_en When set to 1, clocks for USB3 SS core and DMA are gated in P1 state.

Table 721: POWER_GATING_ENABLE_REG Register
Offset: 0x0005004C

Bit	Field	Type/InitVal	Description
Power management register. Please refer to device integration spec, power management chapter for power gating requirement.			
31:4	Reserved_1	RO 0x0	Reserved_1
3	usb2_phy_power_gating_en	RW 0x0	usb2_phy_power_gating_en Reserved
2	ss_phy_power_gating_en	RW 0x0	ss_phy_power_gating_en Reserved.
1	usb3_core_power_gating_en2	RW 0x0	usb3_core_power_gating_en2 When set to1, power can be removed from USB3 core power domain when device is in D3 power state and D3 power state entry is triggered by SoC disabling USB3 subsystem.
0	usb3_core_power_gating_en1	RW 0x0	usb3_core_power_gating_en1 When set to1, power can be removed from USB3 core power domain when device is in D3 power state and D3 power state entry is triggered by U3 entry for SS, or by USB2 Suspend if device is working in USB2 mode.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

A.13.1.5 Core and Endpoint

Table 722: DEV_USB_ADDRESS_REG Register
Offset: 0x00050320

Bit	Field	Type/InitVal	Description
31:7	Reserved_1	RO 0x0	Reserved_1
6:0	dev_reg_usb_dev_addr	RO 0x0	dev_reg_usb_dev_addr Displays current device address after enumeration.

Table 723: FUNCTION_WAKEUP_REG Register
Offset: 0x00050324

Bit	Field	Type/InitVal	Description
31:24	fn_wp_notification_interface_num	RW 0x0	fn_wp_notification_interface_num This field indicates EP interface number to which the notification should be send.
23	fn_wp_notification_enable	RW 0x0	fn_wp_notification_enable This bit, if set, would enable the core to transmit function wakeup notification packets. On transmission of the packet, the hardware will clear this bit.
22:0	Reserved_1	RO 0x0	Reserved_1 Reserved.

Table 724: ENDPOINT_0_CONFIG_REG Register
Offset: 0x00050328

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size_n	RW 0x1	epreg_burst_size For EP0, this field is fixed to 1.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower The field contains lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For EP0, the Max packet Size[10:0] is fixed to 512 bytes.
15:12	Reserved_1	RO 0x0	Reserved_1

Copyright © 2015 Marvell

Table 724: ENDPOINT_0_CONFIG_REG Register (Continued)
Offset: 0x00050328

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit, when set, stalls all the access made to this endpoint. This feature is provided so that USB Function can stall this endpoint if so required. If a Stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a Clear Stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register. For EP0, Stall can only be sent out in data or status stage.
10:8	preg_max_pkt_size_upper	RW 0x2	preg_max_pkt_size_upper These bits contain the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	chg_stage	WO 0x0	chg_stage 0= Normal Operation 1= Wnen Setup packet is received, Controller sets this bit to 1 and sends out ACK with Nump=0. When SW clears this bit, controller sends out ERDY for host to move to data stage or status stage. When Status packet is received, Controller sets this bit to 1 and sends out NRDY to host. When SW clears this bit, controller sends out ERDY to host and subsequently sends out ACK when re-sent Status packet is received which completes the whole control transfer.
6:0	Reserved_2	RO 0x0	Reserved_2

Table 725: OUT_ENDPOINT_CONFIG_REG_1 Register
Offset: 0x0005032C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 725: OUT_ENDPOINT_CONFIG_REG_1 Register (Continued)
Offset: 0x0005032C

Bit	Field	Type/InitVal	Description
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x1	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 1 for this EP.

Copyright © 2015 Marvell

Table 726: OUT_ENDPOINT_CONFIG_REG_2 Register
Offset: 0x00050330

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x3	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 726: OUT_ENDPOINT_CONFIG_REG_2 Register (Continued)
Offset: 0x00050330

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x3	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x2	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 2 for this EP.

Table 727: OUT_ENDPOINT_CONFIG_REG_3 Register
Offset: 0x00050334

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 727: OUT_ENDPOINT_CONFIG_REG_3 Register (Continued)
Offset: 0x00050334

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x1	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x3	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 3 for this EP.

Table 728: OUT_ENDPOINT_CONFIG_REG_4 Register
Offset: 0x00050338

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size field indicates the max number of bursts this EP can transfer.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 728: OUT_ENDPOINT_CONFIG_REG_4 Register (Continued)
Offset: 0x00050338

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 728: OUT_ENDPOINT_CONFIG_REG_4 Register (Continued)
Offset: 0x00050338

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x4	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 4 for this EP.

Table 729: OUT_ENDPOINT_CONFIG_REG_5 Register
Offset: 0x0005033C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size This field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 729: OUT_ENDPOINT_CONFIG_REG_5 Register (Continued)
Offset: 0x0005033C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Copy of Copy of Reserved_3	RO 0x0	Reserved_3
7	bulk_stream_enable	RW 0x0	bulk_stream_enable SW sets this bit to enable bulk stream feature (provided IP is configured with bulk stream feature).
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x5	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 5 for this EP.

Table 730: OUT_ENDPOINT_CONFIG_REG_6 Register
Offset: 0x00050340

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x3	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Table 730: OUT_ENDPOINT_CONFIG_REG_6 Register (Continued)
Offset: 0x00050340

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 730: OUT_ENDPOINT_CONFIG_REG_6 Register (Continued)
Offset: 0x00050340

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x3	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x6	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 6 for this EP.

Table 731: OUT_ENDPOINT_CONFIG_REG_7 Register
Offset: 0x00050344

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 731: OUT_ENDPOINT_CONFIG_REG_7 Register (Continued)
Offset: 0x00050344

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x1	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x7	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 7 for this EP.

Table 732: OUT_ENDPOINT_CONFIG_REG_8 Register
Offset: 0x00050348

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 732: OUT_ENDPOINT_CONFIG_REG_8 Register (Continued)
Offset: 0x00050348

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 732: OUT_ENDPOINT_CONFIG_REG_8 Register (Continued)
Offset: 0x00050348

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x8	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 8 for this EP.

Table 733: OUT_ENDPOINT_CONFIG_REG_9 Register
Offset: 0x0005034C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 733: OUT_ENDPOINT_CONFIG_REG_9 Register (Continued)
Offset: 0x0005034C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x9	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 9 for this EP.

Table 734: OUT_ENDPOINT_CONFIG_REG_10 Register
Offset: 0x00050350

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x3	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell

Table 734: OUT_ENDPOINT_CONFIG_REG_10 Register (Continued)
Offset: 0x00050350

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 734: OUT_ENDPOINT_CONFIG_REG_10 Register (Continued)
Offset: 0x00050350

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x3	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xA	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 10 for this EP.

Table 735: OUT_ENDPOINT_CONFIG_REG_11 Register
Offset: 0x00050354

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 735: OUT_ENDPOINT_CONFIG_REG_11 Register (Continued)
Offset: 0x00050354

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x1	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xB	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 11 for this EP.

Table 736: OUT_ENDPOINT_CONFIG_REG_12 Register
Offset: 0x00050358

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 736: OUT_ENDPOINT_CONFIG_REG_12 Register (Continued)
Offset: 0x00050358

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 736: OUT_ENDPOINT_CONFIG_REG_12 Register (Continued)
Offset: 0x00050358

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xC	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 12 for this EP.

Table 737: OUT_ENDPOINT_CONFIG_REG_13 Register
Offset: 0x0005035C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 737: OUT_ENDPOINT_CONFIG_REG_13 Register (Continued)
Offset: 0x0005035C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xD	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 13 for this EP.

Table 738: OUT_ENDPOINT_CONFIG_REG_14 Register
Offset: 0x00050360

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell

Table 738: OUT_ENDPOINT_CONFIG_REG_14 Register (Continued)
Offset: 0x00050360

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 738: OUT_ENDPOINT_CONFIG_REG_14 Register (Continued)
Offset: 0x00050360

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xE	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 14 for this EP.

Table 739: OUT_ENDPOINT_CONFIG_REG_15 Register
Offset: 0x00050364

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 739: OUT_ENDPOINT_CONFIG_REG_15 Register (Continued)
Offset: 0x00050364

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xF	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 15 for this EP.

Table 740: IN_ENDPOINT_CONFIG_REG_1 Register
Offset: 0x00050368

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell

Table 740: IN_ENDPOINT_CONFIG_REG_1 Register (Continued)
Offset: 0x00050368

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall_in	RW 0x0	ep_stall_in 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 740: IN_ENDPOINT_CONFIG_REG_1 Register (Continued)
Offset: 0x00050368

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x1	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 1 for this EP.

Table 741: IN_ENDPOINT_CONFIG_REG_2 Register
Offset: 0x0005036C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x3	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 741: IN_ENDPOINT_CONFIG_REG_2 Register (Continued)
Offset: 0x0005036C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x3	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x2	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 2 for this EP.

Table 742: IN_ENDPOINT_CONFIG_REG_3 Register
Offset: 0x00050370

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell

Table 742: IN_ENDPOINT_CONFIG_REG_3 Register (Continued)
Offset: 0x00050370

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 742: IN_ENDPOINT_CONFIG_REG_3 Register (Continued)
Offset: 0x00050370

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x1	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x3	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 3 for this EP.

Table 743: IN_ENDPOINT_CONFIG_REG_4 Register
Offset: 0x00050374

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 743: IN_ENDPOINT_CONFIG_REG_4 Register (Continued)
Offset: 0x00050374

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x4	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 4 for this EP.

Table 744: IN_ENDPOINT_CONFIG_REG_5 Register
Offset: 0x00050378

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size This field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 744: IN_ENDPOINT_CONFIG_REG_5 Register (Continued)
Offset: 0x00050378

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 744: IN_ENDPOINT_CONFIG_REG_5 Register (Continued)
Offset: 0x00050378

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x5	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 5 for this EP.

Table 745: IN_ENDPOINT_CONFIG_REG_6 Register
Offset: 0x0005037C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x3	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 745: IN_ENDPOINT_CONFIG_REG_6 Register (Continued)
Offset: 0x0005037C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x3	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x6	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 6 for this EP.

Table 746: IN_ENDPOINT_CONFIG_REG_7 Register
Offset: 0x00050380

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell

Table 746: IN_ENDPOINT_CONFIG_REG_7 Register (Continued)
Offset: 0x00050380

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 746: IN_ENDPOINT_CONFIG_REG_7 Register (Continued)
Offset: 0x00050380

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x1	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x7	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 7 for this EP.

Table 747: IN_ENDPOINT_CONFIG_REG_8 Register
Offset: 0x00050384

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 747: IN_ENDPOINT_CONFIG_REG_8 Register (Continued)
Offset: 0x00050384

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x8	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 8 for this EP.

Table 748: IN_ENDPOINT_CONFIG_REG_9 Register
Offset: 0x00050388

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 748: IN_ENDPOINT_CONFIG_REG_9 Register (Continued)
Offset: 0x00050388

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 748: IN_ENDPOINT_CONFIG_REG_9 Register (Continued)
Offset: 0x00050388

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0x9	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 9 for this EP.

Table 749: IN_ENDPOINT_CONFIG_REG_10 Register
Offset: 0x0005038C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x3	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 749: IN_ENDPOINT_CONFIG_REG_10 Register (Continued)
Offset: 0x0005038C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x3	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xA	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 10 for this EP.

Table 750: IN_ENDPOINT_CONFIG_REG_11 Register
Offset: 0x00050390

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell

Table 750: IN_ENDPOINT_CONFIG_REG_11 Register (Continued)
Offset: 0x00050390

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 750: IN_ENDPOINT_CONFIG_REG_11 Register (Continued)
Offset: 0x00050390

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x1	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xB	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 11 for this EP.

Table 751: IN_ENDPOINT_CONFIG_REG_12 Register
Offset: 0x00050394

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell

Table 751: IN_ENDPOINT_CONFIG_REG_12 Register (Continued)
Offset: 0x00050394

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xC	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 12 for this EP.

Table 752: IN_ENDPOINT_CONFIG_REG_13 Register
Offset: 0x00050398

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 752: IN_ENDPOINT_CONFIG_REG_13 Register (Continued)
Offset: 0x00050398

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Copyright © 2015 Marvell

Table 752: IN_ENDPOINT_CONFIG_REG_13 Register (Continued)
Offset: 0x00050398

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xD	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 13 for this EP.

Table 753: IN_ENDPOINT_CONFIG_REG_14 Register
Offset: 0x0005039C

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 753: IN_ENDPOINT_CONFIG_REG_14 Register (Continued)
Offset: 0x0005039C

Bit	Field	Type/InitVal	Description
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xE	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 14 for this EP.

Table 754: IN_ENDPOINT_CONFIG_REG_15 Register
Offset: 0x000503A0

Bit	Field	Type/InitVal	Description
31:24	epreg_burst_size	RW 0x10	epreg_burst_size The field indicates the max number of bursts this EP can transfer.

Table 754: IN_ENDPOINT_CONFIG_REG_15 Register (Continued)
Offset: 0x000503A0

Bit	Field	Type/InitVal	Description
23:16	epreg_max_pkt_size_lower	RW 0x0	epreg_max_pkt_size_lower These bits contain lower 8 bits of Max Packet Size [10:0] that sets the max number of bytes intended to be transmitted for an IN Packet or received for an OUT packet. For Control EP, the Max packet Size should be fixed to 512 bytes. For other EP, the Max Packet Size is fixed to 1K bytes.
15	Reserved_1	RO 0x0	Reserved_1
14	reset_seq_num	RW 0x0	reset_seq_num This bit is set by software. When this bit is set, the core resets the sequence number. The bit has a self-clearing attribute and the core clears the bit when the sequence number reset is done.
13:12	Reserved_2	RO 0x0	Reserved_2
11	ep_stall	RW 0x0	ep_stall 0 = Do Not STALL this Endpoint 1 = STALL this Endpoint This bit when set STALLs all the access made to this endpoint. This feature is provided so that USB Function can STALL this endpoint if so required. If a stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit permanently, i.e. write to this register cannot take place by USB Function. If a clear stall request to this Endpoint is received via a standard device request command from the host, then EP0 processor will set this bit to 0 and USB Function can read and write to this register.
10:8	epreg_max_pkt_size_upper	RW 0x4	epreg_max_pkt_size_upper This field contains the uppermost 3 bits (10:8) of max_packet_size [10:0].
7	Reserved_3	RO 0x0	Reserved_3
6	endpoint_enable	RW 0x0	endpoint_enable The bit enables/disables this endpoint. 0 = Disable this Endpoint 1 = Enable this Endpoint

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 754: IN_ENDPOINT_CONFIG_REG_15 Register (Continued)
Offset: 0x000503A0

Bit	Field	Type/InitVal	Description
5:4	endpoint_type	RW 0x2	endpoint_type These bits set the type for this Endpoint. 00 = This Endpoint is Control Endpoint 01 = This Endpoint is Isochronous Endpoint 10 = This Endpoint is Bulk Endpoint 11 = This Endpoint is Interrupt Endpoint
3:0	endpoint_num	RW 0xF	endpoint_num This field indicates Endpoint Identifier Number for this endpoint. It is fixed to 15 for this EP.

Table 755: PHY_CFG_REG Register
Offset: 0x000503A4

Bit	Field	Type/InitVal	Description
31:23	Reserved_1	RO 0x0	Reserved_1
22:16	Reserved_2	RO 0x3	Reserved_2
15:12	Reserved_3	RO 0x0	Reserved_3
11	Reserved_4	RO 0x0	Reserved_4
10	usb3_phy_disable_scramble	RW 0x0	usb3_phy_disable_scramble Set to 1 to tell MAC to disable data scrambling.
9	Reserved_5	RO 0x0	Reserved_5
8:5	Reserved_6	RO 0x0	Reserved_6

Copyright © 2015 Marvell

Table 755: PHY_CFG_REG Register (Continued)
Offset: 0x000503A4

Bit	Field	Type/InitVal	Description
4:0	ltssm_st	RO 0x0	ltssm_st 0: START 1: SS_DISABLED 2: SS_INACTIVE_DISC_DETECT 3: SS_INACTIVE_QUIET 4: RX_DETECT_RESET 5: RX_DETECT_ACTIVE 6: RX_DETECT_QUIET 7: POLLING_LFPS 8: POLLING_RXEQ 9: POLLING_ACTIVE 10: POLLING_CONFIGURATION 11: POLLING_IDLE 12: U0_STATE 13: U1_STATE 14: U2_STATE 15: U3_STATE 16: COMPLIANCE 17: RECOVERY_ACTIVE 18: RECOVERY_CONFIGURATION 19: RECOVERY_IDLE 20: LOOPBACK_ACTIVE 21: LOOPBACK_EXIT 22: HOT_RESET_ACTIVE 23: HOT_RESET_EXIT 24: RX_DETECT_LS_MODE, for ssic mode only 25: POLLING_STALL, for ssic mode only 26: SS_INACTIVE 27: SSIC_MPHY_TEST, for ssic mode only 28: SHALLOW_LOOP_BACK

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 756: BUS_INTERVAL_LTM_REG Register
Offset: 0x000503A8

Bit	Field	Type/InitVal	Description
31:16	usb3_bus_int_adjust_value	RW 0x0	usb3_bus_int_adjust_value This value is set by SW. The controller uses this value in the BUS Interval Adjustment notification packet that is transmitted to the host.
15	usb3_bus_int_enable	RW 0x0	usb3_bus_int_enable When this bit is set by SW, the controller transmits Bus Interval Adjustment Notification Packet to the host with the Bus Interval Adjustment Value. This bit is cleared by the core, when this device notification is transmitted.
14:13	Reserved_1	RO 0x0	Reserved_1
12:1	usb3_ltm_belt	RW 0x0	usb3_ltm_belt This value is set by SW. The controller uses this value in the LTM notification packet that is transmitted to the host. This field represents the 12 bits of the BELT value of the LTM notification packet.
0	usb3_ltm_enable	RW 0x0	usb3_ltm_enable When this bit is set by SW, the controller transmits LTM Notification Packet to the host with LTM BELT Value. This bit is cleared by the core, when this device notification is transmitted.

Table 757: ISO_TIME_STAMP_COUNTER_REG Register
Offset: 0x000503AC

Bit	Field	Type/InitVal	Description
31:27	Reserved_1	RO 0x0	Reserved_1
26:0	pl_rcv_isoc_time_stamp	RO 0x0	pl_rcv_isoc_time_stamp The field contains Isochronous Timestamp value from received ITP packet from host.

Table 758: CREDIT_HP_AND_U2_TIMER_REG Register
Offset: 0x000503B0

Bit	Field	Type/InitVal	Description
31:23	Reserved_2	RO 0x0	Reserved_2
22:16	usb3_credit_hp_tto_val	RW 0x46	usb3_credit_hp_tto_val The field contains CREDIT_HP_TIMER timeout value. Updated by SW. Internal use
15	Reserved_1	RO 0x0	Reserved_1

Copyright © 2015 Marvell

Table 758: CREDIT_HP_AND_U2_TIMER_REG Register (Continued)
Offset: 0x000503B0

Bit	Field	Type/InitVal	Description
14	Reserved_0	RO 0x0	Reserved_0
13:8	usb3_pending_hp_tto_val	RW 0x2A	usb3_pending_hp_tto_val The field contains PENDING_HP_TIMER timeout value. Updated by SW. Internal use
7:0	usb3_u2_inactivity_tto_val	RW 0x0	usb3_u2_inactivity_tto_val The field contains U2_INACTIVE_TIMER timeout value. The value placed in this field is the same value that is sent to the hub in a Set Port Feature (PORT_U2_TIMEOUT). Updated by SW.

Table 759: PM_LC_TIMER_TIME_OUT_REG Register
Offset: 0x000503B4

Bit	Field	Type/InitVal	Description
Internal use.			
31:24	Reserved_1	RO 0x0	Reserved_1
23:20	usb3_nolc_timeout_val	RW 0xE	usb3_nolc_timeout_val The field contains NO LINK command within 1.4ms time out value.
19:16	usb3_lup_cto_val	RW 0xA	usb3_lup_cto_val The field contains LUP timer time out value.
15:8	usb3_pm_entry_tto_val	RW 0x54	usb3_pm_entry_tto_val The field contains PM_ENTRY_TIMER time out value.
7:0	usb3_pm_lc_tto_val	RW 0x2A	usb3_pm_lc_tto_val The field contains PM_LC_TIMER time out value.

Table 760: SS_CORE_INTERRUPT_REG Register
Offset: 0x000503B8

Bit	Field	Type/InitVal	Description
31:15	Reserved_1	RO 0x0	Reserved_1
14	setup_rec	WO 0x0	setup_rec 0 = no Setup Packet received from host 1 = Setup Packet received from host
13	trans_rec	WO 0x0	trans_rec 0 = no completion of Transaction 1 = completion of Transaction

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 760: SS_CORE_INTERRUPT_REG Register (Continued)
Offset: 0x000503B8

Bit	Field	Type/InitVal	Description
12	itp_rec	WO 0x0	itp_rec 0 = no ITP Packet received from host 1 = ITP Packet received from host Note: that this bit will NOT be set if a delayed ITP is received.
11	hot_reset_rec	WO 0x0	hot_reset_rec Set to 1 to indicate device receives hot reset.
10:9	Reserved_2	RO 0x0	Reserved_2
8	ltssm_st_change	WO 0x0	ltssm_st_change 1: device entered into new state; 0: no LTSSM state changed;
7	decode_err_8_10b	WO 0x0	decode_err_8_10b 1: 8b/10b decode error detected in data received from PHY; 0: no 8b/10b decode error detected in data received from PHY;
6	disparity_err	WO 0x0	disparity_err 1: disparity error detected in data received from PHY; 0: no disparity error detected in data received from PHY;
5	host_err	RO 0x0	host_err This is set when a ACK TP for any IN endpoint is received with the host error bit set. Software can clear this bit by clearing the bits in EP_HOST_ERROR_STATUS_Register.
4	bus_timeout_error	WO 0x0	bus_timeout_error This bit is set when any bus timeout event is detected on USB bus.
3	phy_rxerr	WO 0x0	phy_rxerr 1: PHY has indicated error (not inclusive of disparity or 8b/10b decode error). 0: No error detected by PHY.
2	crc16_error	WO 0x0	crc16_error 1: CRC16 error detected in header/transaction packet received. 0: No CRC16 error detected.
1	crc5_error	WO 0x0	crc5_error 1: CRC5 error detected in header/transaction packet or in a link command received. 0: No CRC5 error detected.
0	crc32_error	WO 0x0	crc32_error 1: CRC32 error detected in DPP received. 0: No CRC32 error detected.

Copyright © 2015 Marvell

Table 761: SS_CORE_INTERRUPT_EN_REG Register
Offset: 0x000503BC

Bit	Field	Type/InitVal	Description
This interrupt enable register is reset by power-on reset, usb2_bus_reset, vbus off->on, software reset, and internal generated hardware reset (that is SoC disable/enable system, vbus off->on during p3).			
31:15	Reserved_1	RO 0x0	Reserved_1 Set to 1 to enable setup_rec interrupt.
14	setup_rec_en	RW 0x0	setup_rec_en
13	trans_rec_en	RW 0x0	trans_rec_en Set to 1 to enable trans_rec interrupt.
12	itp_rec_en	RW 0x0	itp_rec_en Set to 1 to enable itp_rec interrupt.
11	hot_reset_rec_en	RW 0x0	hot_reset_rec_en Set to 1 to enable hot_reset_rec interrupt.
10:9	Reserved_2	RO 0x0	Reserved_2
8	ltssm_st_change_en	RW 0x0	ltssm_st_change_en Set to 1 to enable ltssm_st_shange interrupt.
7	decode_err_8_10b_en	RW 0x0	decode_err_8_10b_en Set to 1 to enable decode_err_8_10b interrupt.
6	disparity_err_en	RW 0x0	disparity_err_en Set to 1 to enable disparity_err interrupt.
5	host_err_en	RW 0x0	host_err_en Set to 1 to enable host_err interrupt
4	bus_timeout_error_en	RW 0x0	bus_timeout_error_en Set to 1 to enable bus_timeout_err interrupt.
3	phy_rxerr_en	RW 0x0	phy_rxerr_en Set to 1 to enable phy_rxerr interrupt.
2	crc16_error_en	RW 0x0	crc16_error_en Set to 1 to enable crc16_error interrupt.
1	crc5_error_en	RW 0x0	crc5_error_en set to 1 to enable crc5_error interrupt.
0	crc32_error_en	RW 0x0	crc32_error_en Set to 1 to enable crc32_error interrupt.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 762: EP_IN_BINTERVAL_REG_1_2_3 Register
Offset: 0x000503C0

Bit	Field	Type/InitVal	Description
31:24	ep_in_binterval_3	RW 0x2	ep_in_binterval_3 This value specifies interval for isochronous and interrupt endpoint (ep_in_3). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_in_binterval_2	RW 0x2	ep_in_binterval_2 This value specifies interval for isochronous and interrupt endpoint (ep_in_2). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_in_binterval_1	RW 0x2	ep_in_binterval_1 This value specifies interval for isochronous and interrupt endpoint (ep_in_1). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	Reserved_1	RO 0x0	Reserved_1

Table 763: EP_IN_BINTERVAL_REG_4_5_6_7 Register
Offset: 0x000503C4

Bit	Field	Type/InitVal	Description
31:24	ep_in_binterval_7	RW 0x2	ep_in_binterval_7 This value specifies interval for isochronous and interrupt endpoint (ep_in_7). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_in_binterval_6	RW 0x2	ep_in_binterval_6 This value specifies interval for isochronous and interrupt endpoint (ep_in_6). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_in_binterval_5	RW 0x2	ep_in_binterval_5 This value specifies interval for isochronous and interrupt endpoint (ep_in_5). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	ep_in_binterval_4	RW 0x1	ep_in_binterval_4 This value specifies interval for isochronous and interrupt endpoint (ep_in_4). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Table 764: EP_IN_BINTERVAL_REG_8_9_10_11 Register
Offset: 0x000503C8

Bit	Field	Type/InitVal	Description
31:24	ep_in_binterval_11	RW 0x2	ep_in_binterval_11 This value specifies interval for isochronous and interrupt endpoint (ep_in_11). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Copyright © 2015 Marvell

Table 764: EP_IN_BINTERVAL_REG_8_9_10_11 Register (Continued)
Offset: 0x000503C8

Bit	Field	Type/InitVal	Description
23:16	ep_in_binterval_10	RW 0x2	ep_in_binterval_10 This value specifies interval for isochronous and interrupt endpoint (ep_in_10). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_in_binterval_9	RW 0x2	ep_in_binterval_9 This value specifies interval for isochronous and interrupt endpoint (ep_in_9). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	ep_in_binterval_8	RW 0x1	ep_in_binterval_8 This value specifies interval for isochronous and interrupt endpoint (ep_in_8). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Table 765: EP_IN_BINTERVAL_REG_12_13_14_15 Register
Offset: 0x000503CC

Bit	Field	Type/InitVal	Description
31:24	ep_in_binterval_15	RW 0x2	ep_in_binterval_15 This value specifies interval for isochronous and interrupt endpoint (ep_in_15). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_in_binterval_14	RW 0x2	ep_in_binterval_14 This value specifies interval for isochronous and interrupt endpoint (ep_in_14). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_in_binterval_13	RW 0x2	ep_in_binterval_13 This value specifies interval for isochronous and interrupt endpoint (ep_in_13). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	ep_in_binterval_12	RW 0x1	ep_in_binterval_12 This value specifies interval for isochronous and interrupt endpoint (ep_in_12). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Table 766: EP_OUT_BINTERVAL_REG_1_2_3 Register
Offset: 0x000503D0

Bit	Field	Type/InitVal	Description
31:24	ep_out_binterval_3	RW 0x2	ep_out_binterval_3 This value specifies interval for isochronous and interrupt endpoint (ep_out_3). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_out_binterval_2	RW 0x2	ep_out_binterval_2 This value specifies interval for isochronous and interrupt endpoint (ep_out_2). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 766: EP_OUT_BINTERVAL_REG_1_2_3 Register (Continued)
Offset: 0x000503D0

Bit	Field	Type/InitVal	Description
15:8	ep_out_binterval_1	RW 0x2	ep_out_binterval_1 This value specifies interval for isochronous and interrupt endpoint (ep_out_1). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	Reserved_1	RO 0x0	Reserved_1

Table 767: EP_OUT_BINTERVAL_REG_4_5_6_7 Register
Offset: 0x000503D4

Bit	Field	Type/InitVal	Description
31:24	ep_out_binterval_7	RW 0x2	ep_out_binterval_7 This value specifies interval for isochronous and interrupt endpoint (ep_out_7). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_out_binterval_6	RW 0x2	ep_out_binterval_6 This value specifies interval for isochronous and interrupt endpoint (ep_out_6). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_out_binterval_5	RW 0x2	ep_out_binterval_5 This value specifies interval for isochronous and interrupt endpoint (ep_out_5). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	ep_out_binterval_4	RW 0x1	ep_out_binterval_4 This value specifies interval for isochronous and interrupt endpoint (ep_out_4). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Table 768: EP_OUT_BINTERVAL_REG_8_9_10_11 Register
Offset: 0x000503D8

Bit	Field	Type/InitVal	Description
31:24	ep_out_binterval_11	RW 0x2	ep_out_binterval_11 This value specifies interval for isochronous and interrupt endpoint (ep_out_11). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_out_binterval_10	RW 0x2	ep_out_binterval_10 This value specifies interval for isochronous and interrupt endpoint (ep_out_10). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_out_binterval_9	RW 0x2	ep_out_binterval_9 This value specifies interval for isochronous and interrupt endpoint (ep_out_9). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Copyright © 2015 Marvell

Table 768: EP_OUT_BINTERVAL_REG_8_9_10_11 Register (Continued)
Offset: 0x000503D8

Bit	Field	Type/InitVal	Description
7:0	ep_out_binterval_8	RW 0x1	ep_out_binterval_8 This value specifies interval for isochronous and interrupt endpoint (ep_out_8). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Table 769: EP_OUT_BINTERVAL_REG_12_13_14_15 Register
Offset: 0x000503DC

Bit	Field	Type/InitVal	Description
31:24	ep_out_binterval_15	RW 0x2	ep_out_binterval_15 This value specifies interval for isochronous and interrupt endpoint (ep_out_15). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
23:16	ep_out_binterval_14	RW 0x2	ep_out_binterval_14 This value specifies interval for isochronous and interrupt endpoint (ep_out_14). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
15:8	ep_out_binterval_13	RW 0x2	ep_out_binterval_13 This value specifies interval for isochronous and interrupt endpoint (ep_out_13). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.
7:0	ep_out_binterval_12	RW 0x1	ep_out_binterval_12 This value specifies interval for isochronous and interrupt endpoint (ep_out_12). This value shall be in the range from 1 to 16. The interval time is 2^(binterval-1)*125us.

Table 770: EP_HOST_ERROR_STATUS_REG Register
Offset: 0x000503E0

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15	usb3_ep_host_error_15	WO 0x0	usb3_ep_host_error_15 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
14	usb3_ep_host_error_14	WO 0x0	usb3_ep_host_error_14 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
13	usb3_ep_host_error_13	WO 0x0	usb3_ep_host_error_13 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
12	usb3_ep_host_error_12	WO 0x0	usb3_ep_host_error_12 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 770: EP_HOST_ERROR_STATUS_REG Register (Continued)
Offset: 0x000503E0

Bit	Field	Type/InitVal	Description
11	usb3_ep_host_error_11	WO 0x0	usb3_ep_host_error_11 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
10	usb3_ep_host_error_10	WO 0x0	usb3_ep_host_error_10
9	usb3_ep_host_error_9	WO 0x0	usb3_ep_host_error_9 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
8	usb3_ep_host_error_8	WO 0x0	usb3_ep_host_error_8 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
7	usb3_ep_host_error_7	WO 0x0	usb3_ep_host_error_7 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
6	usb3_ep_host_error_6	WO 0x0	usb3_ep_host_error_6 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
5	usb3_ep_host_error_5	WO 0x0	usb3_ep_host_error_5 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
4	usb3_ep_host_error_4	WO 0x0	usb3_ep_host_error_4 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
3	usb3_ep_host_error_3	WO 0x0	usb3_ep_host_error_3 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
2	usb3_ep_host_error_2	WO 0x0	usb3_ep_host_error_2 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
1	usb3_ep_host_error_1	WO 0x0	usb3_ep_host_error_1 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.
0	usb3_ep_host_error_0	WO 0x0	usb3_ep_host_error_0 Bit 0 is for ep0 that received host error; Bit 15 is for ep15 that received host error and so on.

Table 771: UX_EXIT_DETECTION_DELAY_REG Register
Offset: 0x000503E8

Bit	Field	Type/InitVal	Description
Internal use			
31	host_init_ux_dly_en	RW 0x0	host_init_ux_dly_en Host initiated Ux entry, device LFPS detection delay enable. 1= enable; 0 = disable

Copyright © 2015 Marvell

Table 771: UX_EXIT_DETECTION_DELAY_REG Register (Continued)
Offset: 0x000503E8

Bit	Field	Type/InitVal	Description
30	dev_init_ux_dly_en	RW 0x0	dev_init_ux_dly_en Device initiated Ux entry, device LFPS detection delay enable. 1= enable; 0 = disable
29:16	Reserved_1	RO 0x0	Reserved_1
15:0	lfps_detect_dly_cnt	RW 0x14	lfps_detect_dly_cnt Value in 100ns unit, LFPS Ux exit detection delay.

Table 772: TX_TSI_NUM_REG Register
Offset: 0x000503EC

Bit	Field	Type/InitVal	Description
Interanl use			
31:16	min_num_tx_ts1	RW 0x0	min_num_tx_ts1 The minimum number of TS1 orderedset sent by device in Polling.Active and Recovery.Active. Default value is 16'd0 which means by default device will follow spec and start to send TS2 when 8 identical TS1/TS2 is received. Software can modify thisregister to non-zero value to control how many TS1 ordered sets need to be send in Polling.Active/Recovery.Active.
15:0	max_num_tx_ts1	RW 0x3A98	max_num_tx_ts1 Device's maximum TS1 order set number during Recovery.Active. Device will transition to Recover.Configuration if the maximum number of TS1 order sets have been sent and no TS2 response from link partner.

Table 773: START_STATE_DELAY_AND_MIN_TS2_HOT_RESET_REG Register
Offset: 0x000503F0

Bit	Field	Type/InitVal	Description
Internal use.			
31:16	min_num_tx_ts2_hot_reset_assert	RW 0x10	min_num_tx_ts2_hot_reset_assert The minimum number of TS2 ordered set with Reset bit asserted in Hot Reset.Active
15:0	start_state_delay_after_rx_term_on	RW 0x1E	start_state_delay_after_rx_term_on Value in ms, the delay to keep LTSSM state in START state.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 774: LOWPOWER_MANAGEMENT_REG Register
Offset: 0x000503F4

Bit	Field	Type/InitVal	Description
31:4	Reserved_1	RO 0x0	Reserved_1
3	sw_en_init_u2	RW 0x0	sw_en_init_u2 Only when {sw_en_init_u2, sw_en_init_u1}=2'b10, device will initiate LGO_u2 after ep_idle_timer times out.
2	sw_en_init_u1	RW 0x0	sw_en_init_u1 Set 1 means that SW enables device initiate GO_U1 once ep_idle_timer times out.
1	sw_en_reject_u2	RW 0x1	sw_en_reject_u2 Set 1 means that controller rejects host's requirement of entering U2
0	sw_en_reject_u1	RW 0x1	sw_en_reject_u1 Set 1 indicates that controller rejects host's requirement of entering U1.

Table 775: USB3_SETUP_DP_LOW Register
Offset: 0x000503F8

Bit	Field	Type/InitVal	Description
31:0	setup_reg_low	RO 0x0	setup_reg_low The field contains lower 32 bit of current setup packet received from host.

Table 776: USB3_SETUP_DP_HIGH_REG Register
Offset: 0x000503FC

Bit	Field	Type/InitVal	Description
31:0	setup_reg_high	RO 0x0	setup_reg_high The field contains higher 32 bit of current setup packet received from host.

Table 777: USB3_SETUP_CONTROL_REG Register
Offset: 0x00050400

Bit	Field	Type/InitVal	Description
31:1	Reserved_1	RO 0x0	Reserved_1
0	setup_data_fetched	RW 0x1	setup_data_fetched HW clears this bit when 8 bytes setup data is put into USB3_SETUP_DP_LOW/HIGH_REG. SW sets this bit to acknowledge that it has received the setup packet from USB3_SETUP_DP_LOW/HIGH_REG.

Copyright © 2015 Marvell

Table 778: DELAY_U1_CNT_INIT_VALUE_REG Register
Offset: 0x00050404

Bit	Field	Type/InitVal	Description
31:0	delay_u1_cnt_init_value	RW 0x300	delay_u1_cnt_init_value device initiate or accept LGO_U1 if device lgo_u1 condition is met constantly in this window. 1 unit = 1 us. This value can be used when tuning throughput, where one does not want link go to u1 when traffic is heavy.

Table 779: DELAY_U2_CNT_INIT_VALUE_REG Register
Offset: 0x00050408

Bit	Field	Type/InitVal	Description
31:0	delay_u2_cnt_init_value	RW 0x900	delay_u2_cnt_init_value device initiate or accept LGO_U2 if device lgo_u2 condition is met constantly in this window. 1 unit = 1 us. This value can be used when tuning throughput, where one does not want link go to u2 when traffic is heavy.

Table 780: NO_BURST_MODE_REG Register
Offset: 0x0005040C

Bit	Field	Type/InitVal	Description
31:1	reserved	RO 0x0	reserved
0	no_burst_mode	RW 0x0	no_burst_mode if this bit is set, device can only do non-burst transfer. SW needs to set it to 1 if SoC axi_clk is lower than 100MHz).

A.13.1.6 System

Table 781: DMA_GLOBAL_CONFIG_REG Register
Offset: 0x000507D0

Bit	Field	Type/InitVal	Description
31:5	Reserved_1	RO 0x0	Reserved_1
4	dma_iso_control	RW 0x0	dma_iso_control This bit enables special operation for isochronous endpoints. When this bit is set then DMA continuously fetches or writes into memory location pointed by first BD of the ISO endpoint without caring for BUF_RDY bit.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 781: DMA_GLOBAL_CONFIG_REG Register (Continued)
Offset: 0x000507D0

Bit	Field	Type/InitVal	Description
3	dma_intclr_mode	RW 0x0	dma_intclr_mode Interrupt Clear Mode for all Endpoints 0 - DMA Interrupt status bits set, are cleared automatically once the Interrupt status register is read. 1 - DMA Interrupt status bits set are not cleared automatically and host should clear the bits by writing a 1 in the corresponding bit. This is applicable only for Rx/Tx DMA configuration Registers
2	Reserved_2	RO 0x0	Reserved_2
1	dma_run	RW 0x0	dma_run Bit to enable DMA Engine. Software sets this bit to start the DMA Controller. 0 - Disable the DMA Controller. 1 - Enable the DMA Controller. This is a RO bit for the Hardware. Software can reset this bit to stop the DMA operation. Note that any pending request on the Interconnect Master will be completed gracefully. Further requests will not be generated. When the bit is again set, the DMA Controller will continue its operation from where it had halted.
0	Reserved_3	RO 0x0	Reserved_3

Table 782: BULK_STREAMING_ENABLE_REG Register
Offset: 0x000507D4

Bit	Field	Type/InitVal	Description
31	bulk_streaming_enable_in_15	RW 0x0	bulk_streaming_enable_in_15 bulk_streaming_enable for corresponding endpoints. 1 - bulk streaming enable 0 - bulk streaming disable
30	bulk_streaming_enable_in_14	RW 0x0	bulk_streaming_enable_in_14
29	bulk_streaming_enable_in_13	RW 0x0	bulk_streaming_enable_in_13
28	bulk_streaming_enable_in_12	RW 0x0	bulk_streaming_enable_in_12
27	bulk_streaming_enable_in_11	RW 0x0	bulk_streaming_enable_in_11
26	bulk_streaming_enable_in_10	RW 0x0	bulk_streaming_enable_in_10

Copyright © 2015 Marvell

Table 782: BULK_STREAMING_ENABLE_REG Register (Continued)
Offset: 0x000507D4

Bit	Field	Type/InitVal	Description
25	bulk_streaming_enable_in_9	RW 0x0	bulk_streaming_enable_in_9
24	bulk_streaming_enable_in_8	RW 0x0	bulk_streaming_enable_in_8
23	bulk_streaming_enable_in_7	RW 0x0	bulk_streaming_enable_in_7
22	bulk_streaming_enable_in_6	RW 0x0	bulk_streaming_enable_in_6
21	bulk_streaming_enable_in_5	RW 0x0	bulk_streaming_enable_in_5
20	bulk_streaming_enable_in_4	RW 0x0	bulk_streaming_enable_in_4
19	bulk_streaming_enable_in_3	RW 0x0	bulk_streaming_enable_in_3
18	bulk_streaming_enable_in_2	RW 0x0	bulk_streaming_enable_in_2
17	bulk_streaming_enable_in_1	RW 0x0	bulk_streaming_enable_in_1
16	Reserved_1	RO 0x0	Reserved_1
15	bulk_streaming_enable_out_15	RW 0x0	bulk_streaming_enable_out_15
14	bulk_streaming_enable_out_14	RW 0x0	bulk_streaming_enable_out_14
13	bulk_streaming_enable_out_13	RW 0x0	bulk_streaming_enable_out_13
12	bulk_streaming_enable_out_12	RW 0x0	bulk_streaming_enable_out_12
11	bulk_streaming_enable_out_11	RW 0x0	bulk_streaming_enable_out_11
10	bulk_streaming_enable_out_10	RW 0x0	bulk_streaming_enable_out_10
9	bulk_streaming_enable_out_9	RW 0x0	bulk_streaming_enable_out_9
8	bulk_streaming_enable_out_8	RW 0x0	bulk_streaming_enable_out_8
7	bulk_streaming_enable_out_7	RW 0x0	bulk_streaming_enable_out_7
6	bulk_streaming_enable_out_6	RW 0x0	bulk_streaming_enable_out_6

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 782: BULK_STREAMING_ENABLE_REG Register (Continued)
Offset: 0x000507D4

Bit	Field	Type/InitVal	Description
5	bulk_streaming_enable_out_5	RW 0x0	bulk_streaming_enable_out_5
4	bulk_streaming_enable_out_4	RW 0x0	bulk_streaming_enable_out_4
3	bulk_streaming_enable_out_3	RW 0x0	bulk_streaming_enable_out_3
2	bulk_streaming_enable_out_2	RW 0x0	bulk_streaming_enable_out_2
1	bulk_streaming_enable_out_1	RW 0x0	bulk_streaming_enable_out_1
0	Reserved_2	RO 0x0	Reserved_2

Table 783: EP_OUT_RCV_STREAM_ID_REG_1 Register
Offset: 0x000507D8

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_1	RO 0x0	ep_out_rcv_stream_id_1 When bulk stream is enabled, it indicates currently received stream_id for OUT EP1.

Table 784: EP_OUT_RCV_STREAM_ID_REG_2 Register
Offset: 0x000507DC

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_2	RO 0x0	ep_out_rcv_stream_id_2 When bulk stream is enabled, it indicates currently received stream_id for OUT EP2.

Copyright © 2015 Marvell

Table 785: EP_OUT_RCV_STREAM_ID_REG_3 Register
Offset: 0x000507E0

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_3	RO 0x0	ep_out_rcv_stream_id_3 When bulk stream is enabled, it indicates currently received stream_id for OUT EP3.

Table 786: EP_OUT_RCV_STREAM_ID_REG_4 Register
Offset: 0x000507E4

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_4	RO 0x0	ep_out_rcv_stream_id_4 When bulk stream is enabled, it indicates currently received stream_id for OUT EP4.

Table 787: EP_OUT_RCV_STREAM_ID_REG_5 Register
Offset: 0x000507E8

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_5	RO 0x0	ep_out_rcv_stream_id_5 When bulk stream is enabled, it indicates currently received stream_id for OUT EP5.

Table 788: EP_OUT_RCV_STREAM_ID_REG_6 Register
Offset: 0x000507EC

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_6	RO 0x0	ep_out_rcv_stream_id_6 When bulk stream is enabled, it indicates currently received stream_id for OUT EP6.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 789: EP_OUT_RCV_STREAM_ID_REG_7 Register
Offset: 0x000507F0

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_7	RO 0x0	ep_out_rcv_stream_id_7 When bulk stream is enabled, it indicates currently received stream_id for OUT EP7.

Table 790: EP_OUT_RCV_STREAM_ID_REG_8 Register
Offset: 0x000507F4

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_8	RO 0x0	ep_out_rcv_stream_id_8 When bulk stream is enabled, it indicates currently received stream_id for OUT EP8.

Table 791: EP_OUT_RCV_STREAM_ID_REG_9 Register
Offset: 0x000507F8

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_9	RO 0x0	ep_out_rcv_stream_id_9 When bulk stream is enabled, it indicates currently received stream_id for OUT EP9.

Table 792: EP_OUT_RCV_STREAM_ID_REG_10 Register
Offset: 0x000507FC

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_10	RO 0x0	ep_out_rcv_stream_id_10 When bulk stream is enabled, it indicates currently received stream_id for OUT EP10.

Copyright © 2015 Marvell

Table 793: EP_OUT_RCV_STREAM_ID_REG_11 Register
Offset: 0x00050800

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_11	RO 0x0	ep_out_rcv_stream_id_11 When bulk stream is enabled, it indicates currently received stream_id for OUT EP11.

Table 794: EP_OUT_RCV_STREAM_ID_REG_12 Register
Offset: 0x00050804

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_12	RO 0x0	ep_out_rcv_stream_id_12 When bulk stream is enabled, it indicates currently received stream_id for OUT EP12.

Table 795: EP_OUT_RCV_STREAM_ID_REG_13 Register
Offset: 0x00050808

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_13	RO 0x0	ep_out_rcv_stream_id_13 When bulk stream is enabled, it indicates currently received stream_id for OUT EP13.

Table 796: EP_OUT_RCV_STREAM_ID_REG_14 Register
Offset: 0x0005080C

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_14	RO 0x0	ep_out_rcv_stream_id_14 When bulk stream is enabled, it indicates currently received stream_id for OUT EP14.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 797: EP_OUT_RCV_STREAM_ID_REG_15 Register
Offset: 0x00050810

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_out_rcv_stream_id_15	RO 0x0	ep_out_rcv_stream_id_15 When bulk stream is enabled, it indicates currently received stream_id for OUT EP15.

Table 798: EP_IN_RCV_STREAM_ID_REG_1 Register
Offset: 0x00050814

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_1	RO 0x0	ep_in_rcv_stream_id_1 When bulk stream is enabled, it indicates currently received stream_id for IN EP1.

Table 799: EP_IN_RCV_STREAM_ID_REG_2 Register
Offset: 0x00050818

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_2	RO 0x0	ep_in_rcv_stream_id_2 When bulk stream is enabled, it indicates currently received stream_id for IN EP2.

Table 800: EP_IN_RCV_STREAM_ID_REG_3 Register
Offset: 0x0005081C

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_3	RO 0x0	ep_in_rcv_stream_id_3 When bulk stream is enabled, it indicates currently received stream_id for IN EP3.

Copyright © 2015 Marvell

Table 801: EP_IN_RCV_STREAM_ID_REG_4 Register
Offset: 0x00050820

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_4	RO 0x0	ep_in_rcv_stream_id_4 When bulk stream is enabled, it indicates currently received stream_id for IN EP4.

Table 802: EP_IN_RCV_STREAM_ID_REG_5 Register
Offset: 0x00050824

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_5	RO 0x0	ep_in_rcv_stream_id_5 When bulk stream is enabled, it indicates currently received stream_id for IN EP5.

Table 803: EP_IN_RCV_STREAM_ID_REG_6 Register
Offset: 0x00050828

Bit	Field	Type/InitVal	Description
Reserved Register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_6	RO 0x0	ep_in_rcv_stream_id_6 When bulk stream is enabled, it indicates currently received stream_id for IN EP6.

Table 804: EP_IN_RCV_STREAM_ID_REG_7 Register
Offset: 0x0005082C

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_7	RO 0x0	ep_in_rcv_stream_id_7 When bulk stream is enabled, it indicates currently received stream_id for IN EP7.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 805: EP_IN_RCV_STREAM_ID_REG_8 Register
Offset: 0x00050830

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_8	RO 0x0	ep_in_rcv_stream_id_8 When bulk stream is enabled, it indicates currently received stream_id for IN EP8.

Table 806: EP_IN_RCV_STREAM_ID_REG_9 Register
Offset: 0x00050834

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_9	RO 0x0	ep_in_rcv_stream_id_9 When bulk stream is enabled, it indicates currently received stream_id for IN EP9.

Table 807: EP_IN_RCV_STREAM_ID_REG_10 Register
Offset: 0x00050838

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_10	RO 0x0	ep_in_rcv_stream_id_10 When bulk stream is enabled, it indicates currently received stream_id for IN EP10.

Table 808: EP_IN_RCV_STREAM_ID_REG_11 Register
Offset: 0x0005083C

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_11	RO 0x0	ep_in_rcv_stream_id_11 When bulk stream is enabled, it indicates currently received stream_id for IN EP11.

Copyright © 2015 Marvell

Table 809: EP_IN_RCV_STREAM_ID_REG_12 Register
Offset: 0x00050840

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_12	RO 0x0	ep_in_rcv_stream_id_12 When bulk stream is enabled, it indicates currently received stream_id for IN EP12.

Table 810: EP_IN_RCV_STREAM_ID_REG_13 Register
Offset: 0x00050844

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_13	RO 0x0	ep_in_rcv_stream_id_13 When bulk stream is enabled, it indicates currently received stream_id for IN EP13.

Table 811: EP_IN_RCV_STREAM_ID_REG_14 Register
Offset: 0x00050848

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_14	RO 0x0	ep_in_rcv_stream_id_14 When bulk stream is enabled, it indicates currently received stream_id for IN EP14.

Table 812: EP_IN_RCV_STREAM_ID_REG_15 Register
Offset: 0x0005084C

Bit	Field	Type/InitVal	Description
Reserved register			
31:16	Reserved_1	RO 0x0	Reserved_1
15:0	ep_in_rcv_stream_id_15	RO 0x0	ep_in_rcv_stream_id_15 When bulk stream is enabled, it indicates currently received stream_id for IN EP15.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 813: DMA_COMPLETE_SUCCESS_STATUS Register
Offset: 0x00050850

Bit	Field	Type/InitVal	Description
31	ep_in_15_dma_success	WO 0x0	ep_in_15_dma_success When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
30	ep_in_14_dma_success	WO 0x0	ep_in_14_dma_success When set, this bit indicates that Endpoint with Index Number 14 (Not necessarily Endpoint Number 14) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
29	ep_in_13_dma_success	WO 0x0	ep_in_13_dma_success When set, this bit indicates that Endpoint with Index Number 13 (Not necessarily Endpoint Number 13) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
28	ep_in_12_dma_success	WO 0x0	ep_in_12_dma_success When set, this bit indicates that Endpoint with Index Number 12 (Not necessarily Endpoint Number 12) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
27	ep_in_11_dma_success	WO 0x0	ep_in_11_dma_success When set, this bit indicates that Endpoint with Index Number 11 (Not necessarily Endpoint Number 11) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
26	ep_in_10_dma_success	WO 0x0	ep_in_10_dma_success When set, this bit indicates that Endpoint with Index Number 10 (Not necessarily Endpoint Number 10) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell

Table 813: DMA_COMPLETE_SUCCESS_STATUS Register (Continued)
Offset: 0x00050850

Bit	Field	Type/InitVal	Description
25	ep_in_9_dma_success	WO 0x0	ep_in_9_dma_success When set, this bit indicates that Endpoint with Index Number 9 (Not necessarily Endpoint Number 9) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
24	ep_in_8_dma_success	WO 0x0	ep_in_8_dma_success When set, this bit indicates that Endpoint with Index Number 8 (Not necessarily Endpoint Number 8) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
23	ep_in_7_dma_success	WO 0x0	ep_in_7_dma_success When set, this bit indicates that Endpoint with Index Number 7 (Not necessarily Endpoint Number 7) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
22	ep_in_6_dma_success	WO 0x0	ep_in_6_dma_success When set, this bit indicates that Endpoint with Index Number 6 (Not necessarily Endpoint Number 6) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
21	ep_in_5_dma_success	WO 0x0	ep_in_5_dma_success When set, this bit indicates that Endpoint with Index Number 5 (Not necessarily Endpoint Number 5) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
20	ep_in_4_dma_success	WO 0x0	ep_in_4_dma_success When set, this bit indicates that Endpoint with Index Number 4 (Not necessarily Endpoint Number 4) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
19	ep_in_3_dma_success	WO 0x0	ep_in_3_dma_success When set, this bit indicates that Endpoint with Index Number 3 (Not necessarily Endpoint Number 3) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 813: DMA_COMPLETE_SUCCESS_STATUS Register (Continued)
Offset: 0x00050850

Bit	Field	Type/InitVal	Description
18	ep_in_2_dma_success	WO 0x0	ep_in_2_dma_success When set, this bit indicates that Endpoint with Index Number 2 (Not necessarily Endpoint Number 2) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
17	ep_in_1_dma_success	WO 0x0	ep_in_1_dma_success When set, this bit indicates that Endpoint with Index Number 1 (Not necessarily Endpoint Number 1) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
16	ep_in_0_dma_success	WO 0x0	ep_in_0_dma_success When set, this bit indicates that Endpoint with Index Number 0 (Not necessarily Endpoint Number 0) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
15	ep_out_15_dma_success	WO 0x0	ep_out_15_dma_success When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
14	ep_out_14_dma_success	WO 0x0	ep_out_14_dma_success When set, this bit indicates that Endpoint with Index Number 14 (Not necessarily Endpoint Number 14) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
13	ep_out_13_dma_success	WO 0x0	ep_out_13_dma_success When set, this bit indicates that Endpoint with Index Number 13 (Not necessarily Endpoint Number 13) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell

Table 813: DMA_COMPLETE_SUCCESS_STATUS Register (Continued)
Offset: 0x00050850

Bit	Field	Type/InitVal	Description
12	ep_out_12_dma_success	WO 0x0	ep_out_12_dma_success When set, this bit indicates that Endpoint with Index Number 12 (Not necessarily Endpoint Number 12) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
11	ep_out_11_dma_success	WO 0x0	ep_out_11_dma_success When set, this bit indicates that Endpoint with Index Number 11 (Not necessarily Endpoint Number 11) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
10	ep_out_10_dma_success	WO 0x0	ep_out_10_dma_success When set, this bit indicates that Endpoint with Index Number 10 (Not necessarily Endpoint Number 10) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
9	ep_out_9_dma_success	WO 0x0	ep_out_9_dma_success When set, this bit indicates that Endpoint with Index Number 9 (Not necessarily Endpoint Number 9) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
8	ep_out_8_dma_success	WO 0x0	ep_out_8_dma_success When set, this bit indicates that Endpoint with Index Number 8 (Not necessarily Endpoint Number 8) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
7	ep_out_7_dma_success	WO 0x0	ep_out_7_dma_success When set, this bit indicates that Endpoint with Index Number 7 (Not necessarily Endpoint Number 7) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
6	ep_out_6_dma_success	WO 0x0	ep_out_6_dma_success When set, this bit indicates that Endpoint with Index Number 6 (Not necessarily Endpoint Number 6) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 813: DMA_COMPLETE_SUCCESS_STATUS Register (Continued)
Offset: 0x00050850

Bit	Field	Type/InitVal	Description
5	ep_out_5_dma_success	WO 0x0	ep_out_5_dma_success When set, this bit indicates that Endpoint with Index Number 5 (Not necessarily Endpoint Number 5) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
4	ep_out_4_dma_success	WO 0x0	ep_out_4_dma_success When set, this bit indicates that Endpoint with Index Number 4 (Not necessarily Endpoint Number 4) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
3	ep_out_3_dma_success	WO 0x0	ep_out_3_dma_success When set, this bit indicates that Endpoint with Index Number 3 (Not necessarily Endpoint Number 3) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
2	ep_out_2_dma_success	WO 0x0	ep_out_2_dma_success When set, this bit indicates that Endpoint with Index Number 2 (Not necessarily Endpoint Number 2) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
1	ep_out_1_dma_success	WO 0x0	ep_out_1_dma_success When set, this bit indicates that Endpoint with Index Number 1 (Not necessarily Endpoint Number 1) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.
0	ep_out_0_dma_success	WO 0x0	ep_out_0_dma_success When set, this bit indicates that Endpoint with Index Number 0 (Not necessarily Endpoint Number 0) has complteded DMA Read successfully. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell

Table 814: DMA_COMPLETE_ERROR_STATUS Register
Offset: 0x00050854

Bit	Field	Type/InitVal	Description
31	ep_in_15_dma_error	WO 0x0	ep_in_15_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
30	ep_in_14_dma_error	WO 0x0	ep_in_14_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
29	ep_in_13_dma_error	WO 0x0	ep_in_13_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
28	ep_in_12_dma_error	WO 0x0	ep_in_12_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
27	ep_in_11_dma_error	WO 0x0	ep_in_11_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
26	ep_in_10_dma_error	WO 0x0	ep_in_10_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 814: DMA_COMPLETE_ERROR_STATUS Register (Continued)
Offset: 0x00050854

Bit	Field	Type/InitVal	Description
25	ep_in_9_dma_error	WO 0x0	ep_in_9_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
24	ep_in_8_dma_error	WO 0x0	ep_in_8_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
23	ep_in_7_dma_error	WO 0x0	ep_in_7_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
22	ep_in_6_dma_error	WO 0x0	ep_in_6_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
21	ep_in_5_dma_error	WO 0x0	ep_in_5_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
20	ep_in_4_dma_error	WO 0x0	ep_in_4_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell

Table 814: DMA_COMPLETE_ERROR_STATUS Register (Continued)
Offset: 0x00050854

Bit	Field	Type/InitVal	Description
19	ep_in_3_dma_error	WO 0x0	ep_in_3_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
18	ep_in_2_dma_error	WO 0x0	ep_in_2_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
17	ep_in_1_dma_error	WO 0x0	ep_in_1_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
16	ep_in_0_dma_error	WO 0x0	ep_in_0_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Read with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
15	ep_out_15_dma_error	WO 0x0	ep_out_15_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
14	ep_out_14_dma_error	WO 0x0	ep_out_14_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 814: DMA_COMPLETE_ERROR_STATUS Register (Continued)
Offset: 0x00050854

Bit	Field	Type/InitVal	Description
13	ep_out_13_dma_error	WO 0x0	ep_out_13_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
12	ep_out_12_dma_error	WO 0x0	ep_out_12_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
11	ep_out_11_dma_error	WO 0x0	ep_out_11_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
10	ep_out_10_dma_error	WO 0x0	ep_out_10_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
9	ep_out_9_dma_error	WO 0x0	ep_out_9_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
8	ep_out_8_dma_error	WO 0x0	ep_out_8_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Copyright © 2015 Marvell

Table 814: DMA_COMPLETE_ERROR_STATUS Register (Continued)
Offset: 0x00050854

Bit	Field	Type/InitVal	Description
7	ep_out_7_dma_error	WO 0x0	ep_out_7_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
6	ep_out_6_dma_error	WO 0x0	ep_out_6_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
5	ep_out_5_dma_error	WO 0x0	ep_out_5_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
4	ep_out_4_dma_error	WO 0x0	ep_out_4_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
3	ep_out_3_dma_error	WO 0x0	ep_out_3_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
2	ep_out_2_dma_error	WO 0x0	ep_out_2_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 814: DMA_COMPLETE_ERROR_STATUS Register (Continued)
Offset: 0x00050854

Bit	Field	Type/InitVal	Description
1	ep_out_1_dma_error	WO 0x0	ep_out_1_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.
0	ep_out_0_dma_error	WO 0x0	ep_out_0_dma_error When set, this bit indicates that Endpoint with Index Number 15 (Not necessarily Endpoint Number 15) has complteded DMA Write with error. Writing 1 to this location will clear this bit. Writing 0 has no effect.

Table 815: DMA_BD_FETCH_ERROR_STATUS_REG Register
Offset: 0x00050858

Bit	Field	Type/InitVal	Description
31	ep_in_15_dma_bd_fetch_error	WO 0x0	ep_in_15_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
30	ep_in_14_dma_bd_fetch_error	WO 0x0	ep_in_14_dma_bd_fetch_error DMA BD fetch error response status for ep in 14
29	ep_in_13_dma_bd_fetch_error	WO 0x0	ep_in_13_dma_bd_fetch_error DMA BD fetch error response status for ep in 13
28	ep_in_12_dma_bd_fetch_error	WO 0x0	ep_in_12_dma_bd_fetch_error DMA BD fetch error response status for ep in 12
27	ep_in_11_dma_bd_fetch_error	WO 0x0	ep_in_11_dma_bd_fetch_error DMA BD fetch error response status for ep in 11
26	ep_in_10_dma_bd_fetch_error	WO 0x0	ep_in_10_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
25	ep_in_9_dma_bd_fetch_error	WO 0x0	ep_in_9_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
24	ep_in_8_dma_bd_fetch_error	WO 0x0	ep_in_8_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
23	ep_in_7_dma_bd_fetch_error	WO 0x0	ep_in_7_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
22	ep_in_6_dma_bd_fetch_error	WO 0x0	ep_in_6_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
21	ep_in_5_dma_bd_fetch_error	WO 0x0	ep_in_5_dma_bd_fetch_error DMA BD fetch error response status for ep in 15

Copyright © 2015 Marvell

Table 815: DMA_BD_FETCH_ERROR_STATUS_REG Register (Continued)
Offset: 0x00050858

Bit	Field	Type/InitVal	Description
20	ep_in_4_dma_bd_fetch_error	WO 0x0	ep_in_4_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
19	ep_in_3_dma_bd_fetch_error	WO 0x0	ep_in_3_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
18	ep_in_2_dma_bd_fetch_error	WO 0x0	ep_in_2_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
17	ep_in_1_dma_bd_fetch_error	WO 0x0	ep_in_1_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
16	ep_in_0_dma_bd_fetch_error	WO 0x0	ep_in_0_dma_bd_fetch_error DMA BD fetch error response status for ep in 15
15	ep_out_15_dma_bd_fetch_error	WO 0x0	ep_out_15_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
14	ep_out_14_dma_bd_fetch_error	WO 0x0	ep_out_14_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
13	ep_out_13_dma_bd_fetch_error	WO 0x0	ep_out_13_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
12	ep_out_12_dma_bd_fetch_error	WO 0x0	ep_out_12_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
11	ep_out_11_dma_bd_fetch_error	WO 0x0	ep_out_11_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
10	ep_out_10_dma_bd_fetch_error	WO 0x0	ep_out_10_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
9	ep_out_9_dma_bd_fetch_error	WO 0x0	ep_out_9_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
8	ep_out_8_dma_bd_fetch_error	WO 0x0	ep_out_8_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
7	ep_out_7_dma_bd_fetch_error	WO 0x0	ep_out_7_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
6	ep_out_6_dma_bd_fetch_error	WO 0x0	ep_out_6_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
5	ep_out_5_dma_bd_fetch_error	WO 0x0	ep_out_5_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
4	ep_out_4_dma_bd_fetch_error	WO 0x0	ep_out_4_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
3	ep_out_3_dma_bd_fetch_error	WO 0x0	ep_out_3_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
2	ep_out_2_dma_bd_fetch_error	WO 0x0	ep_out_2_dma_bd_fetch_error DMA BD fetch error response status for ep out 15
1	ep_out_1_dma_bd_fetch_error	WO 0x0	ep_out_1_dma_bd_fetch_error DMA BD fetch error response status for ep out 15

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 815: DMA_BD_FETCH_ERROR_STATUS_REG Register (Continued)
Offset: 0x00050858

Bit	Field	Type/InitVal	Description
0	ep_out_0_dma_bd_fetch_error	WO 0x0	ep_out_0_dma_bd_fetch_error DMA BD fetch error response status for ep out 15

Table 816: DMA_BD_FETCH_ERROR_EN_REG Register
Offset: 0x0005085C

Bit	Field	Type/InitVal	Description
31	ep_in_15_dma_bd_fetch_error_en	RW 0x0	ep_in_15_dma_bd_fetch_error_en
30	ep_in_14_dma_bd_fetch_error_en	RW 0x0	ep_in_14_dma_bd_fetch_error_en
29	ep_in_13_dma_bd_fetch_error_en	RW 0x0	ep_in_13_dma_bd_fetch_error_en
28	ep_in_12_dma_bd_fetch_error_en	RW 0x0	ep_in_12_dma_bd_fetch_error_en
27	ep_in_11_dma_bd_fetch_error_en	RW 0x0	ep_in_11_dma_bd_fetch_error_en
26	ep_in_10_dma_bd_fetch_error_en	RW 0x0	ep_in_10_dma_bd_fetch_error_en
25	ep_in_9_dma_bd_fetch_error_en	RW 0x0	ep_in_9_dma_bd_fetch_error_en
24	ep_in_8_dma_bd_fetch_error_en	RW 0x0	ep_in_8_dma_bd_fetch_error_en
23	ep_in_7_dma_bd_fetch_error_en	RW 0x0	ep_in_7_dma_bd_fetch_error_en
22	ep_in_6_dma_bd_fetch_error_en	RW 0x0	ep_in_6_dma_bd_fetch_error_en
21	ep_in_5_dma_bd_fetch_error_en	RW 0x0	ep_in_5_dma_bd_fetch_error_en
20	ep_in_4_dma_bd_fetch_error_en	RW 0x0	ep_in_4_dma_bd_fetch_error_en
19	ep_in_3_dma_bd_fetch_error_en	RW 0x0	ep_in_3_dma_bd_fetch_error_en
18	ep_in_2_dma_bd_fetch_error_en	RW 0x0	ep_in_2_dma_bd_fetch_error_en
17	ep_in_1_dma_bd_fetch_error_en	RW 0x0	ep_in_1_dma_bd_fetch_error_en
16	ep_in_0_dma_bd_fetch_error_en	RW 0x0	ep_in_0_dma_bd_fetch_error_en

Copyright © 2015 Marvell

Table 816: DMA_BD_FETCH_ERROR_EN_REG Register (Continued)
Offset: 0x0005085C

Bit	Field	Type/InitVal	Description
15	ep_out_15_dma_bd_fetch_error_en	RW 0x0	ep_out_15_dma_bd_fetch_error_en
14	ep_out_14_dma_bd_fetch_error_en	RW 0x0	ep_out_14_dma_bd_fetch_error_en
13	ep_out_13_dma_bd_fetch_error_en	RW 0x0	ep_out_13_dma_bd_fetch_error_en
12	ep_out_12_dma_bd_fetch_error_en	RW 0x0	ep_out_12_dma_bd_fetch_error_en
11	ep_out_11_dma_bd_fetch_error_en	RW 0x0	ep_out_11_dma_bd_fetch_error_en
10	ep_out_10_dma_bd_fetch_error_en	RW 0x0	ep_out_10_dma_bd_fetch_error_en
9	ep_out_9_dma_bd_fetch_error_en	RW 0x0	ep_out_9_dma_bd_fetch_error_en
8	ep_out_8_dma_bd_fetch_error_en	RW 0x0	ep_out_8_dma_bd_fetch_error_en
7	ep_out_7_dma_bd_fetch_error_en	RW 0x0	ep_out_7_dma_bd_fetch_error_en
6	ep_out_6_dma_bd_fetch_error_en	RW 0x0	ep_out_6_dma_bd_fetch_error_en
5	ep_out_5_dma_bd_fetch_error_en	RW 0x0	ep_out_5_dma_bd_fetch_error_en
4	ep_out_4_dma_bd_fetch_error_en	RW 0x0	ep_out_4_dma_bd_fetch_error_en
3	ep_out_3_dma_bd_fetch_error_en	RW 0x0	ep_out_3_dma_bd_fetch_error_en
2	ep_out_2_dma_bd_fetch_error_en	RW 0x0	ep_out_2_dma_bd_fetch_error_en
1	ep_out_1_dma_bd_fetch_error_en	RW 0x0	ep_out_1_dma_bd_fetch_error_en
0	ep_out_0_dma_bd_fetch_error_en	RW 0x0	ep_out_0_dma_bd_fetch_error_en

Table 817: DMA_DATA_ERROR_STATUS_REG Register
Offset: 0x00050860

Bit	Field	Type/InitVal	Description
31	ep_in_15_dma_data_error	WO 0x0	ep_in_15_dma_data_error

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 817: DMA_DATA_ERROR_STATUS_REG Register (Continued)
Offset: 0x00050860

Bit	Field	Type/InitVal	Description
30	ep_in_14_dma_data_error	WO 0x0	ep_in_14_dma_data_error
29	ep_in_13_dma_data_error	WO 0x0	ep_in_13_dma_data_error
28	ep_in_12_dma_data_error	WO 0x0	ep_in_12_dma_data_error
27	ep_in_11_dma_data_error	WO 0x0	ep_in_11_dma_data_error
26	ep_in_10_dma_data_error	WO 0x0	ep_in_10_dma_data_error
25	ep_in_9_dma_data_error	WO 0x0	ep_in_9_dma_data_error
24	ep_in_8_dma_data_error	WO 0x0	ep_in_8_dma_data_error
23	ep_in_7_dma_data_error	WO 0x0	ep_in_7_dma_data_error
22	ep_in_6_dma_data_error	WO 0x0	ep_in_6_dma_data_error
21	ep_in_5_dma_data_error	WO 0x0	ep_in_5_dma_data_error
20	ep_in_4_dma_data_error	WO 0x0	ep_in_4_dma_data_error
19	ep_in_3_dma_data_error	WO 0x0	ep_in_3_dma_data_error
18	ep_in_2_dma_data_error	WO 0x0	ep_in_2_dma_data_error
17	ep_in_1_dma_data_error	WO 0x0	ep_in_1_dma_data_error
16	ep_in_0_dma_data_error	WO 0x0	ep_in_0_dma_data_error
15	ep_out_15_dma_data_error	WO 0x0	ep_out_15_dma_data_error
14	ep_out_14_dma_data_error	WO 0x0	ep_out_14_dma_data_error
13	ep_out_13_dma_data_error	WO 0x0	ep_out_13_dma_data_error
12	ep_out_12_dma_data_error	WO 0x0	ep_out_12_dma_data_error
11	ep_out_11_dma_data_error	WO 0x0	ep_out_11_dma_data_error

Copyright © 2015 Marvell

Table 817: DMA_DATA_ERROR_STATUS_REG Register (Continued)
Offset: 0x00050860

Bit	Field	Type/InitVal	Description
10	ep_out_10_dma_data_error	WO 0x0	ep_out_10_dma_data_error
9	ep_out_9_dma_data_error	WO 0x0	ep_out_9_dma_data_error
8	ep_out_8_dma_data_error	WO 0x0	ep_out_8_dma_data_error
7	ep_out_7_dma_data_error	WO 0x0	ep_out_7_dma_data_error
6	ep_out_6_dma_data_error	WO 0x0	ep_out_6_dma_data_error
5	ep_out_5_dma_data_error	WO 0x0	ep_out_5_dma_data_error
4	ep_out_4_dma_data_error	WO 0x0	ep_out_4_dma_data_error
3	ep_out_3_dma_data_error	WO 0x0	ep_out_3_dma_data_error
2	ep_out_2_dma_data_error	WO 0x0	ep_out_2_dma_data_error
1	ep_out_1_dma_data_error	WO 0x0	ep_out_1_dma_data_error
0	ep_out_0_dma_data_error	WO 0x0	ep_out_0_dma_data_error

Table 818: DMA_DATA_ERROR_EN_REG Register
Offset: 0x00050864

Bit	Field	Type/InitVal	Description
31	ep_in_15_dma_data_error_en	RW 0x0	ep_in_15_dma_data_error_en
30	ep_in_14_dma_data_error_en	RW 0x0	ep_in_14_dma_data_error_en
29	ep_in_13_dma_data_error_en	RW 0x0	ep_in_13_dma_data_error_en
28	ep_in_12_dma_data_error_en	RW 0x0	ep_in_12_dma_data_error_en
27	ep_in_11_dma_data_error_en	RW 0x0	ep_in_11_dma_data_error_en
26	ep_in_10_dma_data_error_en	RW 0x0	ep_in_10_dma_data_error_en

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 818: DMA_DATA_ERROR_EN_REG Register (Continued)
Offset: 0x00050864

Bit	Field	Type/InitVal	Description
25	ep_in_9_dma_data_error_en	RW 0x0	ep_in_9_dma_data_error_en
24	ep_in_8_dma_data_error_en	RW 0x0	ep_in_8_dma_data_error_en
23	ep_in_7_dma_data_error_en	RW 0x0	ep_in_7_dma_data_error_en
22	ep_in_6_dma_data_error_en	RW 0x0	ep_in_6_dma_data_error_en
21	ep_in_5_dma_data_error_en	RW 0x0	ep_in_5_dma_data_error_en
20	ep_in_4_dma_data_error_en	RW 0x0	ep_in_4_dma_data_error_en
19	ep_in_3_dma_data_error_en	RW 0x0	ep_in_3_dma_data_error_en
18	ep_in_2_dma_data_error_en	RW 0x0	ep_in_2_dma_data_error_en
17	ep_in_1_dma_data_error_en	RW 0x0	ep_in_1_dma_data_error_en
16	ep_in_0_dma_data_error_en	RW 0x0	ep_in_0_dma_data_error_en
15	ep_out_15_dma_data_error_en	RW 0x0	ep_out_15_dma_data_error_en
14	ep_out_14_dma_data_error_en	RW 0x0	ep_out_14_dma_data_error_en
13	ep_out_13_dma_data_error_en	RW 0x0	ep_out_13_dma_data_error_en
12	ep_out_12_dma_data_error_en	RW 0x0	ep_out_12_dma_data_error_en
11	ep_out_11_dma_data_error_en	RW 0x0	ep_out_11_dma_data_error_en
10	ep_out_10_dma_data_error_en	RW 0x0	ep_out_10_dma_data_error_en
9	ep_out_9_dma_data_error_en	RW 0x0	ep_out_9_dma_data_error_en
8	ep_out_8_dma_data_error_en	RW 0x0	ep_out_8_dma_data_error_en
7	ep_out_7_dma_data_error_en	RW 0x0	ep_out_7_dma_data_error_en
6	ep_out_6_dma_data_error_en	RW 0x0	ep_out_6_dma_data_error_en

Copyright © 2015 Marvell

Table 818: DMA_DATA_ERROR_EN_REG Register (Continued)
Offset: 0x00050864

Bit	Field	Type/InitVal	Description
5	ep_out_5_dma_data_error_en	RW 0x0	ep_out_5_dma_data_error_en
4	ep_out_4_dma_data_error_en	RW 0x0	ep_out_4_dma_data_error_en
3	ep_out_3_dma_data_error_en	RW 0x0	ep_out_3_dma_data_error_en
2	ep_out_2_dma_data_error_en	RW 0x0	ep_out_2_dma_data_error_en
1	ep_out_1_dma_data_error_en	RW 0x0	ep_out_1_dma_data_error_en
0	ep_out_0_dma_data_error_en	RW 0x0	ep_out_0_dma_data_error_en

Table 819: DMA_ERROR_HANDLING_MODE_REG Register
Offset: 0x00050868

Bit	Field	Type/InitVal	Description
Internal use			
31:4	Reserved_1	RO 0x0	Reserved_1
3:2	inband_reset_mode	RW 0x0	inband_reset_mode This field determines the hardware issues a warm/hot reset upon receiving a warm/hot reset detection from MAC: 0 = v0: Issue warm/hot reset to PL/LL/MAC while DMA is not reset 1 = v1: Issue DMA reset; once the DMA has completed all outstanding transactions, hardware resets PL/LL/MAC.
1:0	dma_error_mode	RW 0x0	dma_error_mode 0 = v0: No error handling for AXI error response; SW can enable AXI error response interrupt enable or check the AXI error response status. 1 = v1: Hardware automatically resets DMA error for the endpoint when that endpoint has a BD fetch. BD updates the AXI error response. Hardware asserts the error bit in the BD for the data channel AXI response error. 2 = v2: Hardware automatically resets the DMA error for the endpoint when that endpoint has a BD fetch, BD update, or data channel AXI error response.

Table 820: EP_OUT_RX_DMA_CONFIG_REG_0 Register
Offset: 0x0005086C

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 820: EP_OUT_RX_DMA_CONFIG_REG_0 Register (Continued)
Offset: 0x0005086C

Bit	Field	Type/InitVal	Description
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start This is the door bell of cooresponding ep once SW prepared 1 segment of BD. SW set to 1 for ringing door bell once. HW automatically clear it to 0.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 821: EP_OUT_RX_DMA_CONFIG_REG_1 Register
Offset: 0x00050870

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 822: EP_OUT_RX_DMA_CONFIG_REG_2 Register
Offset: 0x00050874

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Copyright © 2015 Marvell

Table 822: EP_OUT_RX_DMA_CONFIG_REG_2 Register (Continued)
Offset: 0x00050874

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 823: EP_OUT_RX_DMA_CONFIG_REG_3 Register
Offset: 0x00050878

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 824: EP_OUT_RX_DMA_CONFIG_REG_4 Register
Offset: 0x0005087C

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 824: EP_OUT_RX_DMA_CONFIG_REG_4 Register (Continued)
Offset: 0x0005087C

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 825: EP_OUT_RX_DMA_CONFIG_REG_5 Register
Offset: 0x00050880

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 826: EP_OUT_RX_DMA_CONFIG_REG_6 Register
Offset: 0x00050884

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Copyright © 2015 Marvell

Table 826: EP_OUT_RX_DMA_CONFIG_REG_6 Register (Continued)
Offset: 0x00050884

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 827: EP_OUT_RX_DMA_CONFIG_REG_7 Register
Offset: 0x00050888

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 828: EP_OUT_RX_DMA_CONFIG_REG_8 Register
Offset: 0x0005088C

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 828: EP_OUT_RX_DMA_CONFIG_REG_8 Register (Continued)
Offset: 0x0005088C

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 829: EP_OUT_RX_DMA_CONFIG_REG_9 Register
Offset: 0x00050890

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 830: EP_OUT_RX_DMA_CONFIG_REG_10 Register
Offset: 0x00050894

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Copyright © 2015 Marvell

Table 830: EP_OUT_RX_DMA_CONFIG_REG_10 Register (Continued)
Offset: 0x00050894

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 831: EP_OUT_RX_DMA_CONFIG_REG_11 Register
Offset: 0x00050898

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 832: EP_OUT_RX_DMA_CONFIG_REG_12 Register
Offset: 0x0005089C

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 832: EP_OUT_RX_DMA_CONFIG_REG_12 Register (Continued)
Offset: 0x0005089C

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 833: EP_OUT_RX_DMA_CONFIG_REG_13 Register
Offset: 0x000508A0

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 834: EP_OUT_RX_DMA_CONFIG_REG_14 Register
Offset: 0x000508A4

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll

Copyright © 2015 Marvell

Table 834: EP_OUT_RX_DMA_CONFIG_REG_14 Register (Continued)
Offset: 0x000508A4

Bit	Field	Type/InitVal	Description
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 835: EP_OUT_RX_DMA_CONFIG_REG_15 Register
Offset: 0x000508A8

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	rx_dma_poll_interval	RW 0x0	rx_dma_poll_interval
7	rx_dma_poll	RW 0x0	rx_dma_poll
6	rx_dma_start	RW 0x0	rx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 836: EP_IN_TX_DMA_CONFIG_REG_0 Register
Offset: 0x000508AC

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 836: EP_IN_TX_DMA_CONFIG_REG_0 Register (Continued)
Offset: 0x000508AC

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 837: EP_IN_TX_DMA_CONFIG_REG_1 Register
Offset: 0x000508B0

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 838: EP_IN_TX_DMA_CONFIG_REG_2 Register
Offset: 0x000508B4

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Copyright © 2015 Marvell

Table 838: EP_IN_TX_DMA_CONFIG_REG_2 Register (Continued)
Offset: 0x000508B4

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 839: EP_IN_TX_DMA_CONFIG_REG_3 Register
Offset: 0x000508B8

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 840: EP_IN_TX_DMA_CONFIG_REG_4 Register
Offset: 0x000508BC

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 840: EP_IN_TX_DMA_CONFIG_REG_4 Register (Continued)
Offset: 0x000508BC

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 841: EP_IN_TX_DMA_CONFIG_REG_5 Register
Offset: 0x000508C0

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 842: EP_IN_TX_DMA_CONFIG_REG_6 Register
Offset: 0x000508C4

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Copyright © 2015 Marvell

Table 842: EP_IN_TX_DMA_CONFIG_REG_6 Register (Continued)
Offset: 0x000508C4

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 843: EP_IN_TX_DMA_CONFIG_REG_7 Register
Offset: 0x000508C8

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 844: EP_IN_TX_DMA_CONFIG_REG_8 Register
Offset: 0x000508CC

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 844: EP_IN_TX_DMA_CONFIG_REG_8 Register (Continued)
Offset: 0x000508CC

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 845: EP_IN_TX_DMA_CONFIG_REG_9 Register
Offset: 0x000508D0

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 846: EP_IN_TX_DMA_CONFIG_REG_10 Register
Offset: 0x000508D4

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Copyright © 2015 Marvell

Table 846: EP_IN_TX_DMA_CONFIG_REG_10 Register (Continued)
Offset: 0x000508D4

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 847: EP_IN_TX_DMA_CONFIG_REG_11 Register
Offset: 0x000508D8

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 848: EP_IN_TX_DMA_CONFIG_REG_12 Register
Offset: 0x000508DC

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 848: EP_IN_TX_DMA_CONFIG_REG_12 Register (Continued)
Offset: 0x000508DC

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 849: EP_IN_TX_DMA_CONFIG_REG_13 Register
Offset: 0x000508E0

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 850: EP_IN_TX_DMA_CONFIG_REG_14 Register
Offset: 0x000508E4

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll

Copyright © 2015 Marvell

Table 850: EP_IN_TX_DMA_CONFIG_REG_14 Register (Continued)
Offset: 0x000508E4

Bit	Field	Type/InitVal	Description
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 851: EP_IN_TX_DMA_CONFIG_REG_15 Register
Offset: 0x000508E8

Bit	Field	Type/InitVal	Description
31:16	Reserved_1	RO 0x0	Reserved_1
15:8	tx_dma_poll_interval	RW 0x0	tx_dma_poll_interval
7	tx_dma_poll	RW 0x0	tx_dma_poll
6	tx_dma_start	RW 0x0	tx_dma_start SW set this bit after Buffer Descriptor segment is prepared. SW set this bit multiple times for multiple Buffer Descriptor segment. The DMA starts fetching the descriptors. This bit is cleared by DMA controller; SW can only write this bit after it's cleared by DMA controller.
5:0	Reserved_2	RO 0x0	Reserved_2

Table 852: EP_OUT_RX_DMA_START_ADDRESS_REG_0 Register
Offset: 0x000508EC

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the Buffer Descriptor Table in the external memory for EP0 receive data.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 853: EP_OUT_RX_DMA_START_ADDRESS_REG_1 Register
Offset: 0x000508F0

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP1.

Table 854: EP_OUT_RX_DMA_START_ADDRESS_REG_2 Register
Offset: 0x000508F4

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP2.

Table 855: EP_OUT_RX_DMA_START_ADDRESS_REG_3 Register
Offset: 0x000508F8

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP3.

Table 856: EP_OUT_RX_DMA_START_ADDRESS_REG_4 Register
Offset: 0x000508FC

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP4.

Table 857: EP_OUT_RX_DMA_START_ADDRESS_REG_5 Register
Offset: 0x00050900

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP5.

Copyright © 2015 Marvell

Table 858: EP_OUT_RX_DMA_START_ADDRESS_REG_6 Register
Offset: 0x00050904

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP6.

Table 859: EP_OUT_RX_DMA_START_ADDRESS_REG_7 Register
Offset: 0x00050908

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP7.

Table 860: EP_OUT_RX_DMA_START_ADDRESS_REG_8 Register
Offset: 0x0005090C

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP8.

Table 861: EP_OUT_RX_DMA_START_ADDRESS_REG_9 Register
Offset: 0x00050910

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP9.

Table 862: EP_OUT_RX_DMA_START_ADDRESS_REG_10 Register
Offset: 0x00050914

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP10.

Copyright © 2015 Marvell

Table 863: EP_OUT_RX_DMA_START_ADDRESS_REG_11 Register
Offset: 0x00050918

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP11.

Table 864: EP_OUT_RX_DMA_START_ADDRESS_REG_12 Register
Offset: 0x0005091C

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for OUT EP12.

Table 865: EP_OUT_RX_DMA_START_ADDRESS_REG_13 Register
Offset: 0x00050920

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP13.

Table 866: EP_OUT_RX_DMA_START_ADDRESS_REG_14 Register
Offset: 0x00050924

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP14.

Table 867: EP_OUT_RX_DMA_START_ADDRESS_REG_15 Register
Offset: 0x00050928

Bit	Field	Type/InitVal	Description
31:0	csr_odf_start_addr_N	RW 0x0	csr_odf_start_addr_N This field indicates the start address of the RX Buffer Descriptor Table in the external memory for OUT EP15.

Copyright © 2015 Marvell

Table 868: EP_IN_TX_DMA_START_ADDRESS_REG_0 Register
Offset: 0x0005092C

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for EP0 transmit.

Table 869: EP_IN_TX_DMA_START_ADDRESS_REG_1 Register
Offset: 0x00050930

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP1.

Table 870: EP_IN_TX_DMA_START_ADDRESS_REG_2 Register
Offset: 0x00050934

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP2.

Table 871: EP_IN_TX_DMA_START_ADDRESS_REG_3 Register
Offset: 0x00050938

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP3.

Table 872: EP_IN_TX_DMA_START_ADDRESS_REG_4 Register
Offset: 0x0005093C

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP4.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 873: EP_IN_TX_DMA_START_ADDRESS_REG_5 Register
Offset: 0x00050940

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP5.

Table 874: EP_IN_TX_DMA_START_ADDRESS_REG_6 Register
Offset: 0x00050944

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP6.

Table 875: EP_IN_TX_DMA_START_ADDRESS_REG_7 Register
Offset: 0x00050948

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP7.

Table 876: EP_IN_TX_DMA_START_ADDRESS_REG_8 Register
Offset: 0x0005094C

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP8.

Table 877: EP_IN_TX_DMA_START_ADDRESS_REG_9 Register
Offset: 0x00050950

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP9.

Copyright © 2015 Marvell

Table 878: EP_IN_TX_DMA_START_ADDRESS_REG_10 Register
Offset: 0x00050954

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP10.

Table 879: EP_IN_TX_DMA_START_ADDRESS_REG_11 Register
Offset: 0x00050958

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP11.

Table 880: EP_IN_TX_DMA_START_ADDRESS_REG_12 Register
Offset: 0x0005095C

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP12.

Table 881: EP_IN_TX_DMA_START_ADDRESS_REG_13 Register
Offset: 0x00050960

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP13.

Table 882: EP_IN_TX_DMA_START_ADDRESS_REG_14 Register
Offset: 0x00050964

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP14.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 883: EP_IN_TX_DMA_START_ADDRESS_REG_15 Register
Offset: 0x00050968

Bit	Field	Type/InitVal	Description
31:0	csr_idf_start_addr_N	RW 0x0	csr_idf_start_addr_N This field indicates the start address of the TX Buffer Descriptor Table in the external memory for IN EP15.

Table 884: EP_OUT_DMA_ERROR_ADDRESS_REG_0 Register
Offset: 0x0005096C

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For EP0 receive. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 885: EP_OUT_DMA_ERROR_ADDRESS_REG_1 Register
Offset: 0x00050970

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP1. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 886: EP_OUT_DMA_ERROR_ADDRESS_REG_2 Register
Offset: 0x00050974

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP2. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell

Table 887: EP_OUT_DMA_ERROR_ADDRESS_REG_3 Register
Offset: 0x00050978

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP3. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 888: EP_OUT_DMA_ERROR_ADDRESS_REG_4 Register
Offset: 0x0005097C

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP4. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 889: EP_OUT_DMA_ERROR_ADDRESS_REG_5 Register
Offset: 0x00050980

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP5. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 890: EP_OUT_DMA_ERROR_ADDRESS_REG_6 Register
Offset: 0x00050984

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP6. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 891: EP_OUT_DMA_ERROR_ADDRESS_REG_7 Register
Offset: 0x00050988

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP7. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 892: EP_OUT_DMA_ERROR_ADDRESS_REG_8 Register
Offset: 0x0005098C

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP8. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell

Table 893: EP_OUT_DMA_ERROR_ADDRESS_REG_9 Register
Offset: 0x00050990

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP9. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 894: EP_OUT_DMA_ERROR_ADDRESS_REG_10 Register
Offset: 0x00050994

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP10. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 895: EP_OUT_DMA_ERROR_ADDRESS_REG_11 Register
Offset: 0x00050998

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP11. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 896: EP_OUT_DMA_ERROR_ADDRESS_REG_12 Register
Offset: 0x0005099C

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP12. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 897: EP_OUT_DMA_ERROR_ADDRESS_REG_13 Register
Offset: 0x000509A0

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP13. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 898: EP_OUT_DMA_ERROR_ADDRESS_REG_14 Register
Offset: 0x000509A4

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP14. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell

Table 899: EP_OUT_DMA_ERROR_ADDRESS_REG_15 Register
Offset: 0x000509A8

Bit	Field	Type/InitVal	Description
31:0	out_dma_err_addr_o_N	RO 0x0	out_dma_err_addr_o_N For OUT EP15. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 900: EP_IN_DMA_ERROR_ADDRESS_REG_0 Register
Offset: 0x000509AC

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For EP0 transmit. When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 901: EP_IN_DMA_ERROR_ADDRESS_REG_1 Register
Offset: 0x000509B0

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP1 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 902: EP_IN_DMA_ERROR_ADDRESS_REG_2 Register
Offset: 0x000509B4

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP2 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 903: EP_IN_DMA_ERROR_ADDRESS_REG_3 Register
Offset: 0x000509B8

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP3 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 904: EP_IN_DMA_ERROR_ADDRESS_REG_4 Register
Offset: 0x000509BC

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP4 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell

Table 905: EP_IN_DMA_ERROR_ADDRESS_REG_5 Register
Offset: 0x000509C0

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP5 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 906: EP_IN_DMA_ERROR_ADDRESS_REG_6 Register
Offset: 0x000509C4

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP6 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 907: EP_IN_DMA_ERROR_ADDRESS_REG_7 Register
Offset: 0x000509C8

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP7 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 908: EP_IN_DMA_ERROR_ADDRESS_REG_8 Register
Offset: 0x000509CC

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP8 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 909: EP_IN_DMA_ERROR_ADDRESS_REG_9 Register
Offset: 0x000509D0

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP9 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 910: EP_IN_DMA_ERROR_ADDRESS_REG_10 Register
Offset: 0x000509D4

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP10 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Copyright © 2015 Marvell

Table 911: EP_IN_DMA_ERROR_ADDRESS_REG_11 Register
Offset: 0x000509D8

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP11 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 912: EP_IN_DMA_ERROR_ADDRESS_REG_12 Register
Offset: 0x000509DC

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP12 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 913: EP_IN_DMA_ERROR_ADDRESS_REG_13 Register
Offset: 0x000509E0

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP13 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 914: EP_IN_DMA_ERROR_ADDRESS_REG_14 Register
Offset: 0x000509E4

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP14 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 915: EP_IN_DMA_ERROR_ADDRESS_REG_15 Register
Offset: 0x000509E8

Bit	Field	Type/InitVal	Description
31:0	in_dma_err_addr_o_N	RO 0x0	in_dma_err_addr_o_N For IN EP15 When dma completion with error (indicated in DMA_COMPLETE_ERROR_STATUS_REG) or dma fetch error (indicated in DMA_BD_FETCH_ERROR_STATUS_REG) happens, the address that dma tries to access but has error response is logged in this register.

Table 916: DMA_SUSPEND_REG Register
Offset: 0x000509EC

Bit	Field	Type/InitVal	Description
31	csr_in_dma_suspend_i_15	RW 0x0	csr_in_dma_suspend_i_15 DMA suspend function of corresponding endpoint.
30	csr_in_dma_suspend_i_14	RW 0x0	csr_in_dma_suspend_i_14
29	csr_in_dma_suspend_i_13	RW 0x0	csr_in_dma_suspend_i_13
28	csr_in_dma_suspend_i_12	RW 0x0	csr_in_dma_suspend_i_12
27	csr_in_dma_suspend_i_11	RW 0x0	csr_in_dma_suspend_i_11
26	csr_in_dma_suspend_i_10	RW 0x0	csr_in_dma_suspend_i_10
25	csr_in_dma_suspend_i_9	RW 0x0	csr_in_dma_suspend_i_9

Copyright © 2015 Marvell

Table 916: DMA_SUSPEND_REG Register (Continued)
Offset: 0x000509EC

Bit	Field	Type/InitVal	Description
24	csr_in_dma_suspend_i_8	RW 0x0	csr_in_dma_suspend_i_8
23	csr_in_dma_suspend_i_7	RW 0x0	csr_in_dma_suspend_i_7
22	csr_in_dma_suspend_i_6	RW 0x0	csr_in_dma_suspend_i_6
21	csr_in_dma_suspend_i_5	RW 0x0	csr_in_dma_suspend_i_5
20	csr_in_dma_suspend_i_4	RW 0x0	csr_in_dma_suspend_i_4
19	csr_in_dma_suspend_i_3	RW 0x0	csr_in_dma_suspend_i_3
18	csr_in_dma_suspend_i_2	RW 0x0	csr_in_dma_suspend_i_2
17	csr_in_dma_suspend_i_1	RW 0x0	csr_in_dma_suspend_i_1
16	csr_in_dma_suspend_i_0	RW 0x0	csr_in_dma_suspend_i_0
15	csr_out_dma_suspend_i_15	RW 0x0	csr_out_dma_suspend_i_15
14	csr_out_dma_suspend_i_14	RW 0x0	csr_out_dma_suspend_i_14
13	csr_out_dma_suspend_i_13	RW 0x0	csr_out_dma_suspend_i_13
12	csr_out_dma_suspend_i_12	RW 0x0	csr_out_dma_suspend_i_12
11	csr_out_dma_suspend_i_11	RW 0x0	csr_out_dma_suspend_i_11
10	csr_out_dma_suspend_i_10	RW 0x0	csr_out_dma_suspend_i_10
9	csr_out_dma_suspend_i_9	RW 0x0	csr_out_dma_suspend_i_9
8	csr_out_dma_suspend_i_8	RW 0x0	csr_out_dma_suspend_i_8
7	csr_out_dma_suspend_i_7	RW 0x0	csr_out_dma_suspend_i_7
6	csr_out_dma_suspend_i_6	RW 0x0	csr_out_dma_suspend_i_6
5	csr_out_dma_suspend_i_5	RW 0x0	csr_out_dma_suspend_i_5

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 916: DMA_SUSPEND_REG Register (Continued)
Offset: 0x000509EC

Bit	Field	Type/InitVal	Description
24	csr_in_dma_suspend_i_8	RW 0x0	csr_in_dma_suspend_i_8
23	csr_in_dma_suspend_i_7	RW 0x0	csr_in_dma_suspend_i_7
22	csr_in_dma_suspend_i_6	RW 0x0	csr_in_dma_suspend_i_6
21	csr_in_dma_suspend_i_5	RW 0x0	csr_in_dma_suspend_i_5
20	csr_in_dma_suspend_i_4	RW 0x0	csr_in_dma_suspend_i_4
19	csr_in_dma_suspend_i_3	RW 0x0	csr_in_dma_suspend_i_3
18	csr_in_dma_suspend_i_2	RW 0x0	csr_in_dma_suspend_i_2
17	csr_in_dma_suspend_i_1	RW 0x0	csr_in_dma_suspend_i_1
16	csr_in_dma_suspend_i_0	RW 0x0	csr_in_dma_suspend_i_0
15	csr_out_dma_suspend_i_15	RW 0x0	csr_out_dma_suspend_i_15
14	csr_out_dma_suspend_i_14	RW 0x0	csr_out_dma_suspend_i_14
13	csr_out_dma_suspend_i_13	RW 0x0	csr_out_dma_suspend_i_13
12	csr_out_dma_suspend_i_12	RW 0x0	csr_out_dma_suspend_i_12
11	csr_out_dma_suspend_i_11	RW 0x0	csr_out_dma_suspend_i_11
10	csr_out_dma_suspend_i_10	RW 0x0	csr_out_dma_suspend_i_10
9	csr_out_dma_suspend_i_9	RW 0x0	csr_out_dma_suspend_i_9
8	csr_out_dma_suspend_i_8	RW 0x0	csr_out_dma_suspend_i_8
7	csr_out_dma_suspend_i_7	RW 0x0	csr_out_dma_suspend_i_7
6	csr_out_dma_suspend_i_6	RW 0x0	csr_out_dma_suspend_i_6
5	csr_out_dma_suspend_i_5	RW 0x0	csr_out_dma_suspend_i_5

Copyright © 2015 Marvell

Table 916: DMA_SUSPEND_REG Register (Continued)
Offset: 0x000509EC

Bit	Field	Type/InitVal	Description
4	csr_out_dma_suspend_i_4	RW 0x0	csr_out_dma_suspend_i_4
3	csr_out_dma_suspend_i_3	RW 0x0	csr_out_dma_suspend_i_3
2	csr_out_dma_suspend_i_2	RW 0x0	csr_out_dma_suspend_i_2
1	csr_out_dma_suspend_i_1	RW 0x0	csr_out_dma_suspend_i_1
0	csr_out_dma_suspend_i_0	RW 0x0	csr_out_dma_suspend_i_0

Table 917: DMA_SUSPEND_DONE_REG Register
Offset: 0x000509F0

Bit	Field	Type/InitVal	Description
31	csr_in_dma_suspend_done_15	WO 0x0	csr_in_dma_suspend_done_15 After DMA suspend function is finished, HW assert this field. SW to write 1 to clear.
30	csr_in_dma_suspend_done_14	WO 0x0	csr_in_dma_suspend_done_14
29	csr_in_dma_suspend_done_13	WO 0x0	csr_in_dma_suspend_done_13
28	csr_in_dma_suspend_done_12	WO 0x0	csr_in_dma_suspend_done_12
27	csr_in_dma_suspend_done_11	WO 0x0	csr_in_dma_suspend_done_11
26	csr_in_dma_suspend_done_10	WO 0x0	csr_in_dma_suspend_done_10
25	csr_in_dma_suspend_done_9	WO 0x0	csr_in_dma_suspend_done_9
24	csr_in_dma_suspend_done_8	WO 0x0	csr_in_dma_suspend_done_8
23	csr_in_dma_suspend_done_7	WO 0x0	csr_in_dma_suspend_done_7
22	csr_in_dma_suspend_done_6	WO 0x0	csr_in_dma_suspend_done_6
21	csr_in_dma_suspend_done_5	WO 0x0	csr_in_dma_suspend_done_5
20	csr_in_dma_suspend_done_4	WO 0x0	csr_in_dma_suspend_done_4

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 917: DMA_SUSPEND_DONE_REG Register (Continued)
Offset: 0x000509F0

Bit	Field	Type/InitVal	Description
19	csr_in_dma_suspend_done_3	WO 0x0	csr_in_dma_suspend_done_3
18	csr_in_dma_suspend_done_2	WO 0x0	csr_in_dma_suspend_done_2
17	csr_in_dma_suspend_done_1	WO 0x0	csr_in_dma_suspend_done_1
16	csr_in_dma_suspend_done_0	WO 0x0	csr_in_dma_suspend_done_0
15	csr_out_dma_suspend_done_15	WO 0x0	csr_out_dma_suspend_done_15
14	csr_out_dma_suspend_done_14	WO 0x0	csr_out_dma_suspend_done_14
13	csr_out_dma_suspend_done_13	WO 0x0	csr_out_dma_suspend_done_13
12	csr_out_dma_suspend_done_12	WO 0x0	csr_out_dma_suspend_done_12
11	csr_out_dma_suspend_done_11	WO 0x0	csr_out_dma_suspend_done_11
10	csr_out_dma_suspend_done_10	WO 0x0	csr_out_dma_suspend_done_10
9	csr_out_dma_suspend_done_9	WO 0x0	csr_out_dma_suspend_done_9
8	csr_out_dma_suspend_done_8	WO 0x0	csr_out_dma_suspend_done_8
7	csr_out_dma_suspend_done_7	WO 0x0	csr_out_dma_suspend_done_7
6	csr_out_dma_suspend_done_6	WO 0x0	csr_out_dma_suspend_done_6
5	csr_out_dma_suspend_done_5	WO 0x0	csr_out_dma_suspend_done_5
4	csr_out_dma_suspend_done_4	WO 0x0	csr_out_dma_suspend_done_4
3	csr_out_dma_suspend_done_3	WO 0x0	csr_out_dma_suspend_done_3
2	csr_out_dma_suspend_done_2	WO 0x0	csr_out_dma_suspend_done_2
1	csr_out_dma_suspend_done_1	WO 0x0	csr_out_dma_suspend_done_1
0	csr_out_dma_suspend_done_0	WO 0x0	csr_out_dma_suspend_done_0

Copyright © 2015 Marvell

Table 918: DMA_HALT_REG Register
Offset: 0x000509F4

Bit	Field	Type/InitVal	Description
31:16	csr_in_dma_halt_i_N	RW 0x0	csr_in_dma_halt_i_N Software can set a bit to request an IN endpoint to be halted. Each bit corresponds to an IN endpoint. Bit [i] is set to request the IN endpoint with index j to be halted, where j = 31 - i; j varies from 0 to 15. Bits corresponding to unsupported IN endpoints are Reserved. When a bit is set, all DMA transactions from the ccorresponding endpoint are completed gracefully. Any further request received from the USB Host is responded with STALL. The bit encoding is as follows: 0 = v0: Do not halt corresponding IN endpoint. 1 = v1: Halt corresponding IN endpoint.
15:0	csr_out_dma_halt_i_N	RW 0x0	csr_out_dma_halt_i_N Software can set a bit to request an OUT endpoint to be halted. Each bit corresponds to an OUT endpoint. Bit [i] is set to request the OUT endpoint with index i to be halted; i varies from 0 to 15. Bits corresponding to unsupported OUT endpoints are Reserved. When a bit is set, all DMA transactions from the ccorresponding endpoint are completed gracefully. Any further request received from the USB Host is responded with STALL. The bit encoding is as follows: 0 = v0: Do not halt corresponding OUT endpoint. 1 = v1: Halt corresponding OUT endpoint.

Table 919: DMA_HALT_DONE_REG Register
Offset: 0x000509F8

Bit	Field	Type/InitVal	Description
31:16	csr_in_dma_halt_done_N	WO 0x0	csr_in_dma_halt_done_N The bit corresponding to the IN endpoint is set when it has completed the Halt operation. Each bit corresponds to an IN endpoint. Bit [i] - Corresponds to IN endpoint with index j, where j = i - 16, where j varies from 0 to number of supported IN endpoints. Bits corresponding to unsupported endpoints are treated as Reserved bits. The bit encoding for an endpoint is as follows: 0 = v0: Endpoint has not completed Halt operation. 1 = v1: Endpoint has completed Halt operation.

Copyright © 2015 Marvell

Table 919: DMA_HALT_DONE_REG Register (Continued)
Offset: 0x000509F8

Bit	Field	Type/InitVal	Description
15:0	csr_out_dma_halt_done_N	WO 0x0	csr_out_dma_halt_done_N The bit corresponding to the OUT endpoint is set when it has completed the Halt operation. Each bit corresponds to an OUT endpoint. Bit [i] - Corresponds to OUT endpoint with index i, where i varies from 0 to number of supported OUT endpoints. Bits corresponding to unsupported endpoints are treated as Reserved bits. The bit encoding for an endpoint is as follows: 0 = v0: Endpoint has not completed Halt operation. 1 = v1: Endpoint has completed Halt operation.

Table 920: AXI_TIMEOUT_VAL_REG_0 Register
Offset: 0x000509FC

Bit	Field	Type/InitVal	Description
31:0	axi_timeout_0	RW 0x0	axi_timeout_0 This field hold the AXI master read/write response timeout value. Value 0 means that this feature is turned off. The timeout starts to run when an AXI write has written the last beat to the AXI slave and aborts when the write response comes back from the AXI slave. Note: The maximum number of beats that an AXI slave can accept is 16. When the set timeout value is decreased to 0, an interrupt is issued. The timeout counter decrease 1 for every 256 AXI clock.

Table 921: AXI_STATE_REG_1 Register
Offset: 0x00050A00

Bit	Field	Type/InitVal	Description
Intenal register			
31:6	Reserved_1	RO 0x0	Reserved_1
5:4	amst_wr_back_state	RO 0x0	amst_wr_back_state For internal debug.
3:2	Reserved_2	RO 0x0	Reserved_2
1:0	amst_rd_back_state	RO 0x0	amst_rd_back_state For internal debug.

Copyright © 2015 Marvell

Table 922: AXI_STATE_REG_2 Register
Offset: 0x00050A04

Bit	Field	Type/InitVal	Description
Internal register			
31:23	Reserved_1	RO 0x0	Reserved_1
22:20	usb_axi_wr_0_wr_req_state	RO 0x0	usb_axi_wr_0_wr_req_state internal debug field.
19:17	Reserved_2	RO 0x0	Reserved_2
16	usb_axi_wr_0_wr_resp_state	RO 0x0	usb_axi_wr_0_wr_resp_state internal debug field
15	Reserved_3	RO 0x0	Reserved_3
14:12	usb_axi_wr_1_wr_req_state	RO 0x0	usb_axi_wr_1_wr_req_state internal debug field
11:9	Reserved_4	RO 0x0	Reserved_4
8	usb_axi_wr_1_wr_resp_state	RO 0x0	usb_axi_wr_1_wr_resp_state internal debug field
7	Reserved_5	RO 0x0	Reserved_5
6:4	usb_axi_wr_2_wr_req_state	RO 0x0	usb_axi_wr_2_wr_req_state internal debug field.
3:1	Reserve_6	RO 0x0	Reserve_6
0	usb_axi_wr_2_wr_resp_state	RO 0x0	usb_axi_wr_2_wr_resp_state Internal debug field.

Table 923: AXI_STATE_REG_3 Register
Offset: 0x00050A08

Bit	Field	Type/InitVal	Description
31:23	Reserved_1	RO 0x0	Reserved_1
22:20	usb_axi_rd_0_rd_req_state	RO 0x0	usb_axi_rd_0_rd_req_state internal debug field.
19:17	Reserved_2	RO 0x0	Reserved_2
16	usb_axi_rd_0_rd_resp_state	RO 0x0	usb_axi_rd_0_rd_resp_state internal debug field.
15	Reserved_3	RO 0x0	Reserved_3

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 923: AXI_STATE_REG_3 Register (Continued)
Offset: 0x00050A08

Bit	Field	Type/InitVal	Description
14:12	usb_axi_rd_1_rd_req_state	RO 0x0	usb_axi_rd_1_rd_req_state internal debug field.
11:9	Reserved_4	RO 0x0	Reserved_4
8	usb_axi_rd_1_rd_resp_state	RO 0x0	usb_axi_rd_1_rd_resp_state internal debug field.
7	Reserved_5	RO 0x0	Reserved_5
6:4	usb_axi_rd_2_rd_req_state	RO 0x0	usb_axi_rd_2_rd_req_state internal debug field.
3:1	Reserve_6	RO 0x0	Reserve_6
0	usb_axi_rd_2_rd_resp_state	RO 0x0	usb_axi_rd_2_rd_resp_state internal debug field.

Table 924: SS_SYS_INT_REG Register
Offset: 0x00050A0C

Bit	Field	Type/InitVal	Description
31:30	Reserved_1	RO 0x0	Reserved_1
29	axi_data_error_int	RO 0x0	axi_data_error_int This bit is main status bit for dma data error events. This bit is set when there's dma access axi data error in any endpoints. The bit is cleared when the cooresponding endpoint interrupt status register (DMA_DATA_ERROR_STATUS_REG) is cleared by software. Software should treat this field as read-only.
28	axi_bdf_error_int	RO 0x1	axi_bdf_error_int This bit is main status bit for dma BD fetch error events. This bit is set when there's dma BD fetch error in any endpoints. The bit is cleared when the cooresponding endpoint interrupt status register (DMA_BD_FETCH_ERROR_STATUS_REG) is cleared by software. Software should treat this field as read-only.
27	dma_csr_doneq_full	RO 0x0	dma_csr_doneq_full This bit is main status bit for dma DONEQ_FULL error events. This bit is set when there's dma DONEQ_FULL error in any endpoints. The bit is cleared when the cooresponding endpoint interrupt status register (DONEQ_FULL_STATUS_REG) is cleared by software. Software should treat this field as read-only.

Copyright © 2015 Marvell

Table 924: SS_SYS_INT_REG Register (Continued)
Offset: 0x00050A0C

Bit	Field	Type/InitVal	Description
26	bulk_stream_interrupt	RO 0x0	bulk_stream_interrupt This bit is set when any bit in any of the following registers are set: - PRIME_RCV_STATUS_REG - HIMD_RCV_STATUS_REG - STREAM_REJ_STATUS_REG - HOST_FLOWCTRL_STATUS_REG It serves as a main status bit for DMA bulk stream events to allow interrupt generation related to DMA bulk stream activity. SW then searches for the sub-category interrupt status register to check which event exactly occurred. The bit is cleared when the sub-category interrupt status register is cleared by software. Software should treat this field as read-only.
25	dma_rec	RO 0x0	dma_rec This bit is set when any bit in any of the following registers are set: - DMA_Completion_with_Success_Status_Register - DMA_Completion_with_Error_Status_Register It serves as a main status bit for DMA completion events to allow interrupt generation related to DMA completion activity. SW then searches for the sub-category interrupt status register to check which event exactly occurred. The bit is cleared when the sub-category interrupt status register is cleared by software. Software should treat this field as read-only. 0 = v0: No DMA completion event detected by IP 1 = v1: DMA completion event detected by IP
24	halt_all_done	RO 0x0	halt_all_done This bit is set when all endpoints are instructed to halt and all endpoints' DMA gracefully halted. The bit is cleared when the sub-category interrupt status register (DMA_HALT_DONE_REG) is cleared by software. Software should treat this field as read-only.
23:22	Reserved_2	RO 0x0	Reserved_2
21	axi_rd_mas_timeout_2	WO 0x0	axi_rd_mas_timeout_2
20	axi_wr_mas_timeout_2	WO 0x0	axi_wr_mas_timeout_2
19	axi_rd_mas_timeout_1	WO 0x0	axi_rd_mas_timeout_1
18	axi_wr_mas_timeout_1	WO 0x0	axi_wr_mas_timeout_1

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 924: SS_SYS_INT_REG Register (Continued)
Offset: 0x00050A0C

Bit	Field	Type/InitVal	Description
17	axi_rd_mas_timeout_0	WO 0x0	axi_rd_mas_timeout_0
16	axi_wr_mas_timeout_0	WO 0x0	axi_wr_mas_timeout_0
15:14	Reserved_3	RO 0x0	Reserved_3
13	axi_rd_err_2_stat	WO 0x0	axi_rd_err_2_stat
12	axi_rd_err_1_stat	WO 0x0	axi_rd_err_1_stat
11	axi_rd_err_0_stat	WO 0x0	axi_rd_err_0_stat
10	axi_wr_err_2_stat	WO 0x0	axi_wr_err_2_stat
9	axi_wr_err_1_stat	WO 0x0	axi_wr_err_1_stat
8	axi_wr_err_0_stat	WO 0x0	axi_wr_err_0_stat
7:6	Reserved_4	WO 0x0	Reserved_4
5	axi_rd_2_req_zerolen_stat	WO 0x0	axi_rd_2_req_zerolen_stat
4	axi_wr_2_req_zerolen_stat	WO 0x0	axi_wr_2_req_zerolen_stat
3	axi_rd_1_req_zerolen_stat	WO 0x0	axi_rd_1_req_zerolen_stat
2	axi_wr_1_req_zerolen_stat	WO 0x0	axi_wr_1_req_zerolen_stat
1	axi_rd_0_req_zerolen_stat	WO 0x0	axi_rd_0_req_zerolen_stat
0	axi_wr_0_req_zerolen_stat	WO 0x0	axi_wr_0_req_zerolen_stat

Copyright © 2015 Marvell

Table 925: SS_SYS_INT_EN_REG Register
Offset: 0x00050A10

Bit	Field	Type/InitVal	Description
This register provides an interrupt mask for the SS_SYS_INT_REG, defining for each status bit if an interrupt is issued when the corresponding condition occurs. For a description of the conditions for each bit, refer to the description of the SS_SYS_INT_REG.			
31:30	Reserved_1	RO 0x0	Reserved_1
29	axi_data_error_int	RW 0x0	axi_data_error_int_en
28	axi_bdf_error_int	RW 0x0	axi_bdf_error_int_en
27	dma_csr_doneq_full_en	RW 0x0	dma_csr_doneq_full_en
26	bulk_stream_interrupt_en	RW 0x0	bulk_stream_interrupt_en
25	dma_rec_en	RW 0x0	dma_rec_en
24	halt_all_done_en	RW 0x0	halt_all_done_en
23:22	Reserved_2	RO 0x0	Reserved_2
21	axi_rd_mas_timeout_2_en	RW 0x0	axi_rd_mas_timeout_2_en
20	axi_wr_mas_timeout_2_en	RW 0x0	axi_wr_mas_timeout_2_en
19	axi_rd_mas_timeout_1_en	RW 0x0	axi_rd_mas_timeout_1_en
18	axi_wr_mas_timeout_1_en	RW 0x0	axi_wr_mas_timeout_1_en
17	axi_rd_mas_timeout_0_en	RW 0x0	axi_rd_mas_timeout_0_en
16	axi_wr_mas_timeout_0_en	RW 0x0	axi_wr_mas_timeout_0_en
15:14	Reserved_3	RO 0x0	Reserved_3
13	axi_rd_err_2_stat_en	RW 0x0	axi_rd_err_2_stat_en
12	axi_rd_err_1_stat_en	RW 0x0	axi_rd_err_1_stat_en

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 925: SS_SYS_INT_EN_REG Register (Continued)
Offset: 0x00050A10

Bit	Field	Type/InitVal	Description
11	axi_rd_err_0_stat_en	RW 0x0	axi_rd_err_0_stat_en
10	axi_wr_err_2_stat_en	RW 0x0	axi_wr_err_2_stat_en
9	axi_wr_err_1_stat_en	RW 0x0	axi_wr_err_1_stat_en
8	axi_wr_err_0_stat_en	RW 0x0	axi_wr_err_0_stat_en
7:6	Reserved_4	RO 0x0	Reserved_4
5	axi_rd_2_req_zerolen_stat_en	RW 0x0	axi_rd_2_req_zerolen_stat_en
4	axi_wr_2_req_zerolen_stat_en	RW 0x0	axi_wr_2_req_zerolen_stat_en
3	axi_rd_1_req_zerolen_stat_en	RW 0x0	axi_rd_1_req_zerolen_stat_en
2	axi_wr_1_req_zerolen_stat_en	RW 0x0	axi_wr_1_req_zerolen_stat_en
1	axi_rd_0_req_zerolen_stat_en	RW 0x0	axi_rd_0_req_zerolen_stat_en
0	axi_wr_0_req_zerolen_stat_en	RW 0x0	axi_wr_0_req_zerolen_stat_en

Table 926: AXI_ERROR_RESPONSE_CODE_REG_1 Register
Offset: 0x00050A14

Bit	Field	Type/InitVal	Description
Internal register			
31:29	Reserved_1	RO 0x0	Reserved_1
28:20	axi_wr_respID_1	RO 0x0	axi_wr_respID_1 internal debug field
19:18	Reserved_2	RO 0x0	Reserved_2
17:16	axi_wr_err_1	RO 0x0	axi_wr_err_1 internal debug field
15:13	Reserved_3	RO 0x0	Reserved_3
12:4	axi_wr_respID_0	RO 0x0	axi_wr_respID_0 internal debug field

Copyright © 2015 Marvell

Not Approved by Document Control. For Review Only.

Table 926: AXI_ERROR_RESPONSE_CODE_REG_1 Register (Continued)
Offset: 0x00050A14

Bit	Field	Type/InitVal	Description
3:2	Reserved_4	RO 0x0	Reserved_4
1:0	axi_wr_err_0	RO 0x0	axi_wr_err_0 internal debug field.

Table 927: AXI_ERROR_REPONSE_CODE_REG_2 Register
Offset: 0x00050A18

Bit	Field	Type/InitVal	Description
Internal register			
31:13	Reserved_1	RO 0x0	Reserved_1
12:4	axi_wr_respID_2	RO 0x0	axi_wr_respID_2 internal debug field
3:2	Reserved_2	RO 0x0	Reserved_2
1:0	axi_wr_err_2	RO 0x0	axi_wr_err_2 internal debug field

Table 928: AXI_ERROR_RESPONSE_CODE_REG_3 Register
Offset: 0x00050A1C

Bit	Field	Type/InitVal	Description
Internal register			
31:29	Reserved_1	RO 0x0	Reserved_1
28:20	axi_rd_respID_1	RO 0x0	axi_rd_respID_1 internal debug field.
19:18	Reserved_2	RO 0x0	Reserved_2
17:16	axi_rd_err_1	RO 0x0	axi_rd_err_1 internal debug field
15:13	Reserved_3	RO 0x0	Reserved_3
12:4	axi_rd_respID_0	RO 0x0	axi_rd_respID_0 internal debug field
3:2	Reserved_4	RO 0x0	Reserved_4
1:0	axi_rd_err_0	RO 0x0	axi_rd_err_0 internal debug field.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 929: AXI_ERROR_RESPONSE_CODE_REG_4 Register
Offset: 0x00050A20

Bit	Field	Type/InitVal	Description
Internal register			
31:13	Reserved_1	RO 0x0	Reserved_1
12:4	axi_rd_respID_2	RO 0x0	axi_rd_respID_2 internal debug field
3:2	Reserved_2	RO 0x0	Reserved_2
1:0	axi_rd_err_2	RO 0x0	axi_rd_err_2 internal debug field

Table 930: DMA_STATE_REG_1 Register
Offset: 0x00050A24

Bit	Field	Type/InitVal	Description
This register is for internal usage, debug only.			
31:24	dbg_write_avail_buf_internal	RO 0x0	dbg_write_avail_buf_internal
23:5	Reserved_1	RO 0x0	Reserved_1
4	dbg_dma_b2b_ack	RW 0x0	dbg_dma_b2b_ack Safe guard register in case newly added feature of "handling b2b ack without cycle limitation" does not work as expected.
3:0	dbg_dma_mux	RW 0x0	dbg_dma_mux Set this field to the endpoint number to be monitored. That cooresponding endpoint's internal signals can be read out from DMA_STATE_REG_2 and DMA_STATE_REG_3.

Table 931: DMA_STATE_REG_2 Register
Offset: 0x00050A28

Bit	Field	Type/InitVal	Description
Internal register. For debug only.			
31:30	Reserved_1	RO 0x0	Reserved_1
29:24	dbg_dma_out_chache_empty	RO 0x1F	dbg_dma_out_cache_empty these 6 bits are for debug monitoring purpose. bd_cache_full,fin_cache_empty,resp_cache_empty,req_cache_empty,pni_cache_empty,chk_cache_empty.
23	dbg_ep_write_fifo_full	RO 0x0	dbg_ep_write_fifo_full

Copyright © 2015 Marvell

Table 931: DMA_STATE_REG_2 Register (Continued)
Offset: 0x00050A28

Bit	Field	Type/InitVal	Description
22:18	dbg_write_avail_buf_cnt	RO 0x0	dbg_write_avail_buf_cnt
17:16	dbg_dma_out_cache_ptr	RW 0x0	dbg_dma_out_cache_ptr Set this field to the cache pointer to be monitored. That cooresponding internal pointer values can be read out from field dbg_st_dma_out_cache in this register.
15:0	dbg_st_dma_out_cache	RO 0x0	dbg_st_dma_out_cache Display BD cache pointers at Check stage, PNPI stage, Data write stage, Data write done stage, and finish stage.

Table 932: DMA_STATE_REG_3 Register
Offset: 0x00050A2C

Bit	Field	Type/InitVal	Description
This register is for internal usage, debug only.			
31:30	Reserved_1	RO 0x0	Reserved_1
29:24	dbg_dma_in_cache_empty	RO 0x1F	dbg_dma_in_cache_empty these 6 bits are for debug monitoring purpose. bd_cache_full,fin_cache_empty,resp_cache_empty,req_cache_empty,pni_cache_empty,chk_cache_empty.
23	dbg_ep_read_fifo_empty	RO 0x1	dbg_ep_read_fifo_empty
22:18	dbg_read_avail_buf_cnt	RO 0x0	dbg_read_avail_buf_cnt
17:16	dbg_dma_in_cache_ptr	RW 0x0	dbg_dma_in_cache_ptr Set this field to the cache pointer to be monitored. That cooresponding internal pointer values can be read out from field dbg_st_dma_in_cache in this register.
15:0	dbg_st_dma_in_cache	RO 0x0	dbg_st_dma_in_cache Display BD cache pointers at Check stage, PNPI stage, Data request stage, ACK stage.and finish stage.

Table 933: DMA_STATE_REG_4 Register
Offset: 0x00050A30

Bit	Field	Type/InitVal	Description
This register is for internal usage, debug only.			
31:0	Reserved_1	RO 0x0	Reserved_1

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 934: DMA_STATE_REG_5 Register
Offset: 0x00050A34

Bit	Field	Type/InitVal	Description
This register is for internal usage, debug only.			
31:0	Reserved_1	RO 0x0	Reserved_1

Table 935: SEGMENT_COUNTER_REG_0 Register
Offset: 0x00050A38

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N for EP0 transmit. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N for EP0 receive. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 936: SEGMENT_COUNTER_REG_1 Register
Offset: 0x00050A3C

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N for IN EP1. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N for OUT EP1. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Copyright © 2015 Marvell

Table 937: SEGMENT_COUNTER_REG_2 Register
Offset: 0x00050A40

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP2. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP2. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 938: SEGMENT_COUNTER_REG_3 Register
Offset: 0x00050A44

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP3. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP3. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 939: SEGMENT_COUNTER_REG_4 Register
Offset: 0x00050A48

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP4. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP4. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 940: SEGMENT_COUNTER_REG_5 Register
Offset: 0x00050A4C

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP5. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP5. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 941: SEGMENT_COUNTER_REG_6 Register
Offset: 0x00050A50

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP6. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP6. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 942: SEGMENT_COUNTER_REG_7 Register
Offset: 0x00050A54

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP7. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP7. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Copyright © 2015 Marvell

Table 943: SEGMENT_COUNTER_REG_8 Register
Offset: 0x00050A58

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP 8. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP8. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 944: SEGMENT_COUNTER_REG_9 Register
Offset: 0x00050A5C

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN Ep9. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP9. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 945: SEGMENT_COUNTER_REG_10 Register
Offset: 0x00050A60

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP10. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP10. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 946: SEGMENT_COUNTER_REG_11 Register
Offset: 0x00050A64

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP11. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP11. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 947: SEGMENT_COUNTER_REG_12 Register
Offset: 0x00050A68

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For EP EP12. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP12. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 948: SEGMENT_COUNTER_REG_13 Register
Offset: 0x00050A6C

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP13. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP13. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Copyright © 2015 Marvell

Table 949: SEGMENT_COUNTER_REG_14 Register
Offset: 0x00050A70

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For EP EP14. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP14. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 950: SEGMENT_COUNTER_REG_15 Register
Offset: 0x00050A74

Bit	Field	Type/InitVal	Description
31:16	in_seg_cnt_N	RO 0x0	in_seg_cnt_N For IN EP15. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.
15:0	out_seg_cnt_N	RO 0x0	out_seg_cnt_N For OUT EP15. This value indicates how many segments SW has handed over to HW. One segment is a set of BDs where the last BD has nxt_bd_rdy=0.

Table 951: EP_IN_DONEQ_START_ADDRESS_REG_0 Register
Offset: 0x00050A78

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for corresponding EP0 transmit.

Table 952: EP_IN_DONEQ_START_ADDRESS_REG_1 Register
Offset: 0x00050A7C

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP1.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 953: EP_IN_DONEQ_START_ADDRESS_REG_2 Register
Offset: 0x00050A80

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP2.

Table 954: EP_IN_DONEQ_START_ADDRESS_REG_3 Register
Offset: 0x00050A84

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP3.

Table 955: EP_IN_DONEQ_START_ADDRESS_REG_4 Register
Offset: 0x00050A88

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP4.

Table 956: EP_IN_DONEQ_START_ADDRESS_REG_5 Register
Offset: 0x00050A8C

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP5.

Table 957: EP_IN_DONEQ_START_ADDRESS_REG_6 Register
Offset: 0x00050A90

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP6.

Table 958: EP_IN_DONEQ_START_ADDRESS_REG_7 Register
Offset: 0x00050A94

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP7.

Copyright © 2015 Marvell

Table 959: EP_IN_DONEQ_START_ADDRESS_REG_8 Register
Offset: 0x00050A98

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP8.

Table 960: EP_IN_DONEQ_START_ADDRESS_REG_9 Register
Offset: 0x00050A9C

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP9.

Table 961: EP_IN_DONEQ_START_ADDRESS_REG_10 Register
Offset: 0x00050AA0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP10.

Table 962: EP_IN_DONEQ_START_ADDRESS_REG_11 Register
Offset: 0x00050AA4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP11.

Table 963: EP_IN_DONEQ_START_ADDRESS_REG_12 Register
Offset: 0x00050AA8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP12.

Table 964: EP_IN_DONEQ_START_ADDRESS_REG_13 Register
Offset: 0x00050AAC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP13.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 965: EP_IN_DONEQ_START_ADDRESS_REG_14 Register
Offset: 0x00050AB0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP14.

Table 966: EP_IN_DONEQ_START_ADDRESS_REG_15 Register
Offset: 0x00050AB4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_start_addr_N	RW 0x0	in_doneq_start_addr_N SW configures DoneQ start address for IN EP15.

Table 967: EP_IN_DONEQ_END_ADDRESS_REG_0 Register
Offset: 0x00050AB8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for EP0 transmit.

Table 968: EP_IN_DONEQ_END_ADDRESS_REG_1 Register
Offset: 0x00050ABC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP1.

Table 969: EP_IN_DONEQ_END_ADDRESS_REG_2 Register
Offset: 0x00050AC0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP2.

Table 970: EP_IN_DONEQ_END_ADDRESS_REG_3 Register
Offset: 0x00050AC4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP3.

Copyright © 2015 Marvell

Table 971: EP_IN_DONEQ_END_ADDRESS_REG_4 Register
Offset: 0x00050AC8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP4.

Table 972: EP_IN_DONEQ_END_ADDRESS_REG_5 Register
Offset: 0x00050ACC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP5.

Table 973: EP_IN_DONEQ_END_ADDRESS_REG_6 Register
Offset: 0x00050AD0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP6.

Table 974: EP_IN_DONEQ_END_ADDRESS_REG_7 Register
Offset: 0x00050AD4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP7.

Table 975: EP_IN_DONEQ_END_ADDRESS_REG_8 Register
Offset: 0x00050AD8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP8.

Table 976: EP_IN_DONEQ_END_ADDRESS_REG_9 Register
Offset: 0x00050ADC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP9.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 977: EP_IN_DONEQ_END_ADDRESS_REG_10 Register
Offset: 0x00050AE0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP10.

Table 978: EP_IN_DONEQ_END_ADDRESS_REG_11 Register
Offset: 0x00050AE4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP11.

Table 979: EP_IN_DONEQ_END_ADDRESS_REG_12 Register
Offset: 0x00050AE8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP12.

Table 980: EP_IN_DONEQ_END_ADDRESS_REG_13 Register
Offset: 0x00050AEC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP13.

Table 981: EP_IN_DONEQ_END_ADDRESS_REG_14 Register
Offset: 0x00050AF0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP14.

Table 982: EP_IN_DONEQ_END_ADDRESS_REG_15 Register
Offset: 0x00050AF4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_end_addr_N	RW 0x0	in_doneq_end_addr_N SW configures DoneQ end address for IN EP15.

Copyright © 2015 Marvell

Table 983: EP_OUT_DONEQ_START_ADDRESS_REG_0 Register
Offset: 0x00050AF8

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for EP0 receive.

Table 984: EP_OUT_DONEQ_START_ADDRESS_REG_1 Register
Offset: 0x00050AFC

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP1.

Table 985: EP_OUT_DONEQ_START_ADDRESS_REG_2 Register
Offset: 0x00050B00

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP2.

Table 986: EP_OUT_DONEQ_START_ADDRESS_REG_3 Register
Offset: 0x00050B04

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP3.

Table 987: EP_OUT_DONEQ_START_ADDRESS_REG_4 Register
Offset: 0x00050B08

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP4.

Table 988: EP_OUT_DONEQ_START_ADDRESS_REG_5 Register
Offset: 0x00050B0C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP5.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 989: EP_OUT_DONEQ_START_ADDRESS_REG_6 Register
Offset: 0x00050B10

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP6.

Table 990: EP_OUT_DONEQ_START_ADDRESS_REG_7 Register
Offset: 0x00050B14

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP7.

Table 991: EP_OUT_DONEQ_START_ADDRESS_REG_8 Register
Offset: 0x00050B18

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP8.

Table 992: EP_OUT_DONEQ_START_ADDRESS_REG_9 Register
Offset: 0x00050B1C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP9.

Table 993: EP_OUT_DONEQ_START_ADDRESS_REG_10 Register
Offset: 0x00050B20

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP10.

Table 994: EP_OUT_DONEQ_START_ADDRESS_REG_11 Register
Offset: 0x00050B24

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP11.

Copyright © 2015 Marvell

Table 995: EP_OUT_DONEQ_START_ADDRESS_REG_12 Register
Offset: 0x00050B28

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP12.

Table 996: EP_OUT_DONEQ_START_ADDRESS_REG_13 Register
Offset: 0x00050B2C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP13.

Table 997: EP_OUT_DONEQ_START_ADDRESS_REG_14 Register
Offset: 0x00050B30

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP14.

Table 998: EP_OUT_DONEQ_START_ADDRESS_REG_15 Register
Offset: 0x00050B34

Bit	Field	Type/InitVal	Description
31:0	out_doneq_start_addr_N	RW 0x0	out_doneq_start_addr_N SW configures DoneQ start address for OUT EP15.

Table 999: EP_OUT_DONEQ_END_ADDRESS_REG_0 Register
Offset: 0x00050B38

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for EP0 receive.

Table 1000: EP_OUT_DONEQ_END_ADDRESS_REG_1 Register
Offset: 0x00050B3C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP1.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1001: EP_OUT_DONEQ_END_ADDRESS_REG_2 Register
Offset: 0x00050B40

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP2.

Table 1002: EP_OUT_DONEQ_END_ADDRESS_REG_3 Register
Offset: 0x00050B44

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP3.

Table 1003: EP_OUT_DONEQ_END_ADDRESS_REG_4 Register
Offset: 0x00050B48

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP4.

Table 1004: EP_OUT_DONEQ_END_ADDRESS_REG_5 Register
Offset: 0x00050B4C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP5.

Table 1005: EP_OUT_DONEQ_END_ADDRESS_REG_6 Register
Offset: 0x00050B50

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP6.

Table 1006: EP_OUT_DONEQ_END_ADDRESS_REG_7 Register
Offset: 0x00050B54

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP7.

Copyright © 2015 Marvell

Table 1007: EP_OUT_DONEQ_END_ADDRESS_REG_8 Register
Offset: 0x00050B58

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP8.

Table 1008: EP_OUT_DONEQ_END_ADDRESS_REG_9 Register
Offset: 0x00050B5C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP9.

Table 1009: EP_OUT_DONEQ_END_ADDRESS_REG_10 Register
Offset: 0x00050B60

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP10.

Table 1010: EP_OUT_DONEQ_END_ADDRESS_REG_11 Register
Offset: 0x00050B64

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP11.

Table 1011: EP_OUT_DONEQ_END_ADDRESS_REG_12 Register
Offset: 0x00050B68

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP12.

Table 1012: EP_OUT_DONEQ_END_ADDRESS_REG_13 Register
Offset: 0x00050B6C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP13.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1013: EP_OUT_DONEQ_END_ADDRESS_REG_14 Register
Offset: 0x00050B70

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP14.

Table 1014: EP_OUT_DONEQ_END_ADDRESS_REG_15 Register
Offset: 0x00050B74

Bit	Field	Type/InitVal	Description
31:0	out_doneq_end_addr_N	RW 0x0	out_doneq_end_addr_N SW configures DoneQ end address for OUT EP15.

Table 1015: EP_IN_DONEQ_WRITE_POINTER_REG_0 Register
Offset: 0x00050B78

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for EP0 transmit, pointing to next to-be-update DoneQ address.

Table 1016: EP_IN_DONEQ_WRITE_POINTER_REG_1 Register
Offset: 0x00050B7C

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP1, pointing to next to-be-update DoneQ address.

Table 1017: EP_IN_DONEQ_WRITE_POINTER_REG_2 Register
Offset: 0x00050B80

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP2, pointing to next to-be-update DoneQ address.

Copyright © 2015 Marvell

Table 1018: EP_IN_DONEQ_WRITE_POINTER_REG_3 Register
Offset: 0x00050B84

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP3, pointing to next to-be-update DoneQ address.

Table 1019: EP_IN_DONEQ_WRITE_POINTER_REG_4 Register
Offset: 0x00050B88

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP4, pointing to next to-be-update DoneQ address.

Table 1020: EP_IN_DONEQ_WRITE_POINTER_REG_5 Register
Offset: 0x00050B8C

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP5, pointing to next to-be-update DoneQ address.

Table 1021: EP_IN_DONEQ_WRITE_POINTER_REG_6 Register
Offset: 0x00050B90

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP6, pointing to next to-be-update DoneQ address.

Table 1022: EP_IN_DONEQ_WRITE_POINTER_REG_7 Register
Offset: 0x00050B94

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP7, pointing to next to-be-update DoneQ address.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1023: EP_IN_DONEQ_WRITE_POINTER_REG_8 Register
Offset: 0x00050B98

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP8, pointing to next to-be-update DoneQ address.

Table 1024: EP_IN_DONEQ_WRITE_POINTER_REG_9 Register
Offset: 0x00050B9C

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP9, pointing to next to-be-update DoneQ address.

Table 1025: EP_IN_DONEQ_WRITE_POINTER_REG_10 Register
Offset: 0x00050BA0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP10, pointing to next to-be-update DoneQ address.

Table 1026: EP_IN_DONEQ_WRITE_POINTER_REG_11 Register
Offset: 0x00050BA4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP11, pointing to next to-be-update DoneQ address.

Table 1027: EP_IN_DONEQ_WRITE_POINTER_REG_12 Register
Offset: 0x00050BA8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP12, pointing to next to-be-update DoneQ address.

Copyright © 2015 Marvell

Table 1028: EP_IN_DONEQ_WRITE_POINTER_REG_13 Register
Offset: 0x00050BAC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP13, pointing to next to-be-update DoneQ address.

Table 1029: EP_IN_DONEQ_WRITE_POINTER_REG_14 Register
Offset: 0x00050BB0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP14, pointing to next to-be-update DoneQ address.

Table 1030: EP_IN_DONEQ_WRITE_POINTER_REG_15 Register
Offset: 0x00050BB4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_wptr_N	RO 0x0	in_doneq_wptr_N Controller updates DoneQ write pointer for IN EP15, pointing to next to-be-update DoneQ address.

Table 1031: EP_IN_DONEQ_READ_POINTER_REG_0 Register
Offset: 0x00050BB8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For EP0 transmit. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1032: EP_IN_DONEQ_READ_POINTER_REG_1 Register
Offset: 0x00050BBC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP1. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1033: EP_IN_DONEQ_READ_POINTER_REG_2 Register
Offset: 0x00050BC0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP2. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1034: EP_IN_DONEQ_READ_POINTER_REG_3 Register
Offset: 0x00050BC4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP3. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1035: EP_IN_DONEQ_READ_POINTER_REG_4 Register
Offset: 0x00050BC8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP4. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1036: EP_IN_DONEQ_READ_POINTER_REG_5 Register
Offset: 0x00050BCC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP5. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell

Table 1037: EP_IN_DONEQ_READ_POINTER_REG_6 Register
Offset: 0x00050BD0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP6. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1038: EP_IN_DONEQ_READ_POINTER_REG_7 Register
Offset: 0x00050BD4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP7. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1039: EP_IN_DONEQ_READ_POINTER_REG_8 Register
Offset: 0x00050BD8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP8. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1040: EP_IN_DONEQ_READ_POINTER_REG_9 Register
Offset: 0x00050BDC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP9. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1041: EP_IN_DONEQ_READ_POINTER_REG_10 Register
Offset: 0x00050BE0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP10. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1042: EP_IN_DONEQ_READ_POINTER_REG_11 Register
Offset: 0x00050BE4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP11. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1043: EP_IN_DONEQ_READ_POINTER_REG_12 Register
Offset: 0x00050BE8

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP12. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1044: EP_IN_DONEQ_READ_POINTER_REG_13 Register
Offset: 0x00050BEC

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP13. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell

Table 1045: EP_IN_DONEQ_READ_POINTER_REG_14 Register
Offset: 0x00050BF0

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP14. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1046: EP_IN_DONEQ_READ_POINTER_REG_15 Register
Offset: 0x00050BF4

Bit	Field	Type/InitVal	Description
31:0	in_doneq_rptr_N	RW 0x0	in_doneq_rptr_N For IN EP15. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1047: EP_OUT_DONEQ_WRITE_POINTER_REG_0 Register
Offset: 0x00050BF8

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for EP0 receive, pointing to next to-be-update DoneQ address.

Table 1048: EP_OUT_DONEQ_WRITE_POINTER_REG_1 Register
Offset: 0x00050BFC

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP1, pointing to next to-be-update DoneQ address.

Table 1049: EP_OUT_DONEQ_WRITE_POINTER_REG_2 Register
Offset: 0x00050C00

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP2, pointing to next to-be-update DoneQ address.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1050: EP_OUT_DONEQ_WRITE_POINTER_REG_3 Register
Offset: 0x00050C04

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP3, pointing to next to-be-update DoneQ address.

Table 1051: EP_OUT_DONEQ_WRITE_POINTER_REG_4 Register
Offset: 0x00050C08

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP4, pointing to next to-be-update DoneQ address.

Table 1052: EP_OUT_DONEQ_WRITE_POINTER_REG_5 Register
Offset: 0x00050C0C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP5, pointing to next to-be-update DoneQ address.

Table 1053: EP_OUT_DONEQ_WRITE_POINTER_REG_6 Register
Offset: 0x00050C10

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP6, pointing to next to-be-update DoneQ address.

Table 1054: EP_OUT_DONEQ_WRITE_POINTER_REG_7 Register
Offset: 0x00050C14

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP7, pointing to next to-be-update DoneQ address.

Copyright © 2015 Marvell

Table 1055: EP_OUT_DONEQ_WRITE_POINTER_REG_8 Register
Offset: 0x00050C18

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP8, pointing to next to-be-update DoneQ address.

Table 1056: EP_OUT_DONEQ_WRITE_POINTER_REG_9 Register
Offset: 0x00050C1C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP9, pointing to next to-be-update DoneQ address.

Table 1057: EP_OUT_DONEQ_WRITE_POINTER_REG_10 Register
Offset: 0x00050C20

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP10, pointing to next to-be-update DoneQ address.

Table 1058: EP_OUT_DONEQ_WRITE_POINTER_REG_11 Register
Offset: 0x00050C24

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP11, pointing to next to-be-update DoneQ address.

Table 1059: EP_OUT_DONEQ_WRITE_POINTER_REG_12 Register
Offset: 0x00050C28

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP12, pointing to next to-be-update DoneQ address.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1060: EP_OUT_DONEQ_WRITE_POINTER_REG_13 Register
Offset: 0x00050C2C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP13, pointing to next to-be-update DoneQ address.

Table 1061: EP_OUT_DONEQ_WRITE_POINTER_REG_14 Register
Offset: 0x00050C30

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP14, pointing to next to-be-update DoneQ address.

Table 1062: EP_OUT_DONEQ_WRITE_POINTER_REG_15 Register
Offset: 0x00050C34

Bit	Field	Type/InitVal	Description
31:0	out_doneq_wptr_N	RO 0x0	out_doneq_wptr_N Controller updates DoneQ write pointer for OUT EP15, pointing to next to-be-update DoneQ address.

Table 1063: EP_OUT_DONEQ_READ_POINTER_REG_0 Register
Offset: 0x00050C38

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For EP0 receive. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1064: EP_OUT_DONEQ_READ_POINTER_REG_1 Register
Offset: 0x00050C3C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP1. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell

Table 1065: EP_OUT_DONEQ_READ_POINTER_REG_2 Register
Offset: 0x00050C40

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP2. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1066: EP_OUT_DONEQ_READ_POINTER_REG_3 Register
Offset: 0x00050C44

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP3. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1067: EP_OUT_DONEQ_READ_POINTER_REG_4 Register
Offset: 0x00050C48

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP4. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1068: EP_OUT_DONEQ_READ_POINTER_REG_5 Register
Offset: 0x00050C4C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP5. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 1069: EP_OUT_DONEQ_READ_POINTER_REG_6 Register
Offset: 0x00050C50

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP6. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1070: EP_OUT_DONEQ_READ_POINTER_REG_7 Register
Offset: 0x00050C54

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP7. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1071: EP_OUT_DONEQ_READ_POINTER_REG_8 Register
Offset: 0x00050C58

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP8. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1072: EP_OUT_DONEQ_READ_POINTER_REG_9 Register
Offset: 0x00050C5C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP9. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell

Table 1073: EP_OUT_DONEQ_READ_POINTER_REG_10 Register
Offset: 0x00050C60

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP10. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1074: EP_OUT_DONEQ_READ_POINTER_REG_11 Register
Offset: 0x00050C64

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP11. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1075: EP_OUT_DONEQ_READ_POINTER_REG_12 Register
Offset: 0x00050C68

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP12. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1076: EP_OUT_DONEQ_READ_POINTER_REG_13 Register
Offset: 0x00050C6C

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP13. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1077: EP_OUT_DONEQ_READ_POINTER_REG_14 Register
Offset: 0x00050C70

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP14. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1078: EP_OUT_DONEQ_READ_POINTER_REG_15 Register
Offset: 0x00050C74

Bit	Field	Type/InitVal	Description
31:0	out_doneq_rptr_N	RW 0x0	out_doneq_rptr_N For OUT EP15. SW is responsible to move this Read Pointer after it reads out DONEQ data structure.

Table 1079: DONEQ_FULL_STATUS_REG Register
Offset: 0x00050C78

Bit	Field	Type/InitVal	Description
31	in_doneq_full_15	RO 0x1	in_doneq_full_15 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
30	in_doneq_full_14	RO 0x1	in_doneq_full_14 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
29	in_doneq_full_13	RO 0x1	in_doneq_full_13 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.

Copyright © 2015 Marvell

Table 1079: DONEQ_FULL_STATUS_REG Register (Continued)
Offset: 0x00050C78

Bit	Field	Type/InitVal	Description
28	in_doneq_full_12	RO 0x1	in_doneq_full_12 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
27	in_doneq_full_11	RO 0x1	in_doneq_full_11 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
26	in_doneq_full_10	RO 0x1	in_doneq_full_10 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
25	in_doneq_full_9	RO 0x1	in_doneq_full_9 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
24	in_doneq_full_8	RO 0x1	in_doneq_full_8 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
23	in_doneq_full_7	RO 0x1	in_doneq_full_7 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1079: DONEQ_FULL_STATUS_REG Register (Continued)
Offset: 0x00050C78

Bit	Field	Type/InitVal	Description
22	in_doneq_full_6	RO 0x1	in_doneq_full_6 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
21	in_doneq_full_5	RO 0x1	in_doneq_full_5 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
20	in_doneq_full_4	RO 0x1	in_doneq_full_4 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
19	in_doneq_full_3	RO 0x1	in_doneq_full_3 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
18	in_doneq_full_2	RO 0x1	in_doneq_full_2 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
17	in_doneq_full_1	RO 0x1	in_doneq_full_1 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.

Copyright © 2015 Marvell

Table 1079: DONEQ_FULL_STATUS_REG Register (Continued)
Offset: 0x00050C78

Bit	Field	Type/InitVal	Description
16	in_doneq_full_0	RO 0x1	in_doneq_full_0 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
15	out_doneq_full_15	RO 0x1	out_doneq_full_15 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
14	out_doneq_full_14	RO 0x1	out_doneq_full_14 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
13	out_doneq_full_13	RO 0x1	out_doneq_full_13 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
12	out_doneq_full_12	RO 0x1	out_doneq_full_12 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
11	out_doneq_full_11	RO 0x1	out_doneq_full_11 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1079: DONEQ_FULL_STATUS_REG Register (Continued)
Offset: 0x00050C78

Bit	Field	Type/InitVal	Description
10	out_doneq_full_10	RO 0x1	out_doneq_full_10 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
9	out_doneq_full_9	RO 0x1	out_doneq_full_9 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
8	out_doneq_full_8	RO 0x1	out_doneq_full_8 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
7	out_doneq_full_7	RO 0x1	out_doneq_full_7 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
6	out_doneq_full_6	RO 0x1	out_doneq_full_6 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
5	out_doneq_full_5	RO 0x1	out_doneq_full_5 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.

Copyright © 2015 Marvell

Table 1079: DONEQ_FULL_STATUS_REG Register (Continued)
Offset: 0x00050C78

Bit	Field	Type/InitVal	Description
4	out_doneq_full_4	RO 0x1	out_doneq_full_4 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
3	out_doneq_full_3	RO 0x1	out_doneq_full_3 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
2	out_doneq_full_2	RO 0x1	out_doneq_full_2 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
1	out_doneq_full_1	RO 0x1	out_doneq_full_1 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.
0	out_doneq_full_0	RO 0x1	out_doneq_full_0 Read out value 1 indicates that DONEQ is full. Normally DONEQ full is an error condition. Suggest SW to reset DMA if this exception happens. However, this field's reset value is 1 since DONEQ start address and end address reset values are all 0. The read out value will become 0 after SW correctly configures DONEQ start address and end address.

Table 1080: ABORT_REQ_STATUS_REG Register
Offset: 0x00050C7C

Bit	Field	Type/InitVal	Description
31	in_abort_req_15	RW 0x0	in_abort_req_15 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1080: ABORT_REQ_STATUS_REG Register (Continued)
Offset: 0x00050C7C

Bit	Field	Type/InitVal	Description
30	in_abort_req_14	RW 0x0	in_abort_req_14 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
29	in_abort_req_13	RW 0x0	in_abort_req_13 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
28	in_abort_req_12	RW 0x0	in_abort_req_12 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
27	in_abort_req_11	RW 0x0	in_abort_req_11 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
26	in_abort_req_10	RW 0x0	in_abort_req_10 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
25	in_abort_req_9	RW 0x0	in_abort_req_9 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
24	in_abort_req_8	RW 0x0	in_abort_req_8 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
23	in_abort_req_7	RW 0x0	in_abort_req_7 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
22	in_abort_req_6	RW 0x0	in_abort_req_6 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
21	in_abort_req_5	RW 0x0	in_abort_req_5 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
20	in_abort_req_4	RW 0x0	in_abort_req_4 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
19	in_abort_req_3	RW 0x0	in_abort_req_3 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.

Copyright © 2015 Marvell

Table 1080: ABORT_REQ_STATUS_REG Register (Continued)
Offset: 0x00050C7C

Bit	Field	Type/InitVal	Description
18	in_abort_req_2	RW 0x0	in_abort_req_2 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
17	in_abort_req_1	RW 0x0	in_abort_req_1 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
16	Reserved_1	RO 0x0	Reserved_1
15	out_abort_req_15	RW 0x0	out_abort_req_15 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
14	out_abort_req_14	RW 0x0	out_abort_req_14 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
13	out_abort_req_13	RW 0x0	out_abort_req_13 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
12	out_abort_req_12	RW 0x0	out_abort_req_12 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
11	out_abort_req_11	RW 0x0	out_abort_req_11 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
10	out_abort_req_10	RW 0x0	out_abort_req_10 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
9	out_abort_req_9	RW 0x0	out_abort_req_9 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
8	out_abort_req_8	RW 0x0	out_abort_req_8 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
7	out_abort_req_7	RW 0x0	out_abort_req_7 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1080: ABORT_REQ_STATUS_REG Register (Continued)
Offset: 0x00050C7C

Bit	Field	Type/InitVal	Description
6	out_abort_req_6	RW 0x0	out_abort_req_6 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
5	out_abort_req_5	RW 0x0	out_abort_req_5 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
4	out_abort_req_4	RW 0x0	out_abort_req_4 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
3	out_abort_req_3	RW 0x0	out_abort_req_3 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
2	out_abort_req_2	RW 0x0	out_abort_req_2 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
1	out_abort_req_1	RW 0x0	out_abort_req_1 SW asserts this bit to request abort, the stream IDs to be aborted are referred to "Abort_ID_Register_N". HW de-asserts this bit once it finishes HW abort sequence.
0	Reserved_2	RO 0x0	Reserved_2

Table 1081: ABORT_ID_REG_1 Register
Offset: 0x00050C84

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP1. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP1. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Copyright © 2015 Marvell

Table 1082: ABORT_ID_REG_2 Register
Offset: 0x00050C88

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP2. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP2. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1083: ABORT_ID_REG_3 Register
Offset: 0x00050C8C

Bit	Field	Type/InitVal	Description
Used when bulk stream is enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP3. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP3. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1084: ABORT_ID_REG_4 Register
Offset: 0x00050C90

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP4. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1084: ABORT_ID_REG_4 Register (Continued)
Offset: 0x00050C90

Bit	Field	Type/InitVal	Description
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP4. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1085: ABORT_ID_REG_5 Register
Offset: 0x00050C94

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP5. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP5. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1086: ABORT_ID_REG_6 Register
Offset: 0x00050C98

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP6. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP6. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Copyright © 2015 Marvell

Table 1087: ABORT_ID_REG_7 Register
Offset: 0x00050C9C

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP7. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP7. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1088: ABORT_ID_REG_8 Register
Offset: 0x00050CA0

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP8. Each IN can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP8. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1089: ABORT_ID_REG_9 Register
Offset: 0x00050CA4

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP9. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1089: ABORT_ID_REG_9 Register (Continued)
Offset: 0x00050CA4

Bit	Field	Type/InitVal	Description
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP9. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1090: ABORT_ID_REG_10 Register
Offset: 0x00050CA8

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP10. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP10. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1091: ABORT_ID_REG_11 Register
Offset: 0x00050CAC

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP11. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP11. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Copyright © 2015 Marvell

Table 1092: ABORT_ID_REG_12 Register
Offset: 0x00050CB0

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP12. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP12. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1093: ABORT_ID_REG_13 Register
Offset: 0x00050CB4

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP13. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP13. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1094: ABORT_ID_REG_14 Register
Offset: 0x00050CB8

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP14. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1094: ABORT_ID_REG_14 Register (Continued)
Offset: 0x00050CB8

Bit	Field	Type/InitVal	Description
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP14. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1095: ABORT_ID_REG_15 Register
Offset: 0x00050CBC

Bit	Field	Type/InitVal	Description
Reserved when bulk stream is not enabled.			
31:16	in_abort_ID_N	RW 0x0	in_abort_ID_N For IN EP15. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of in_abort_ID_N.
15:0	out_abort_ID_N	RW 0x0	out_abort_ID_N For OUT EP15. Each EP can support Max 256 Stream_ID. Contains 2 encoded stream IDs to be aborted set by SW. If encoded stream ID to be abort is 0, it has to be place in LSB of out_abort_ID_N.

Table 1096: FLUSHED_BD_COUNTER_REG_1 Register
Offset: 0x00050CC0

Bit	Field	Type/InitVal	Description
Internal use.			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Copyright © 2015 Marvell

Table 1097: FLUSHED_BD_COUNTER_REG_2 Register
Offset: 0x00050CC4

Bit	Field	Type/InitVal	Description
Internal use.			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1098: FLUSHED_BD_COUNTER_REG_3 Register
Offset: 0x00050CC8

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1099: FLUSHED_BD_COUNTER_REG_4 Register
Offset: 0x00050CCC

Bit	Field	Type/InitVal	Description
Internal use.			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1100: FLUSHED_BD_COUNTER_REG_5 Register
Offset: 0x00050CD0

Bit	Field	Type/InitVal	Description
Internal use.			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1101: FLUSHED_BD_COUNTER_REG_6 Register
Offset: 0x00050CD4

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1102: FLUSHED_BD_COUNTER_REG_7 Register
Offset: 0x00050CD8

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Copyright © 2015 Marvell

Table 1103: FLUSHED_BD_COUNTER_REG_8 Register
Offset: 0x00050CDC

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1104: FLUSHED_BD_COUNTER_REG_9 Register
Offset: 0x00050CE0

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1105: FLUSHED_BD_COUNTER_REG_10 Register
Offset: 0x00050CE4

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1106: FLUSHED_BD_COUNTER_REG_11 Register
Offset: 0x00050CE8

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1107: FLUSHED_BD_COUNTER_REG_12 Register
Offset: 0x00050CEC

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1108: FLUSHED_BD_COUNTER_REG_13 Register
Offset: 0x00050CF0

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Copyright © 2015 Marvell

Table 1109: FLUSHED_BD_COUNTER_REG_14 Register
Offset: 0x00050CF4

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1110: FLUSHED_BD_COUNTER_REG_15 Register
Offset: 0x00050CF8

Bit	Field	Type/InitVal	Description
Internal use			
31:16	in_flushed_cnt_N	RO 0x0	in_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.
15:0	out_flushed_cnt_N	RO 0x0	out_flushed_cnt_N Indicate how many BD is returned to NRDYQ because of host stream reject or host stream flow control.

Table 1111: EP_IN_FLUSHED_SID_BITMAP_REG_1 Register
Offset: 0x00050CFC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1112: EP_IN_FLUSHED_SID_BITMAP_REG_2 Register
Offset: 0x00050D1C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1113: EP_IN_FLUSHED_SID_BITMAP_REG_3 Register
Offset: 0x00050D3C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1114: EP_IN_FLUSHED_SID_BITMAP_REG_4 Register
Offset: 0x00050D5C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1115: EP_IN_FLUSHED_SID_BITMAP_REG_5 Register
Offset: 0x00050D7C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1116: EP_IN_FLUSHED_SID_BITMAP_REG_6 Register
Offset: 0x00050D9C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1117: EP_IN_FLUSHED_SID_BITMAP_REG_7 Register
Offset: 0x00050DBC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Copyright © 2015 Marvell

Table 1118: EP_IN_FLUSHED_SID_BITMAP_REG_8 Register
Offset: 0x00050DDC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1119: EP_IN_FLUSHED_SID_BITMAP_REG_9 Register
Offset: 0x00050DFC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1120: EP_IN_FLUSHED_SID_BITMAP_REG_10 Register
Offset: 0x00050E1C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1121: EP_IN_FLUSHED_SID_BITMAP_REG_11 Register
Offset: 0x00050E3C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1122: EP_IN_FLUSHED_SID_BITMAP_REG_12 Register
Offset: 0x00050E5C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Copyright © 2015 Marvell

Table 1123: EP_IN_FLUSHED_SID_BITMAP_REG_13 Register
Offset: 0x00050E7C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1124: EP_IN_FLUSHED_SID_BITMAP_REG_14 Register
Offset: 0x00050E9C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1125: EP_IN_FLUSHED_SID_BITMAP_REG_15 Register
Offset: 0x00050EBC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	in_flushed_sid_bitmap_N	RO 0x0	in_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1126: EP_OUT_FLUSHED_SID_BITMAP_REG_1 Register
Offset: 0x00050EDC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1127: EP_OUT_FLUSHED_SID_BITMAP_REG_2 Register
Offset: 0x00050EFC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Copyright © 2015 Marvell

Table 1128: EP_OUT_FLUSHED_SID_BITMAP_REG_3 Register
Offset: 0x00050F1C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1129: EP_OUT_FLUSHED_SID_BITMAP_REG_4 Register
Offset: 0x00050F3C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1130: EP_OUT_FLUSHED_SID_BITMAP_REG_5 Register
Offset: 0x00050F5C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1131: EP_OUT_FLUSHED_SID_BITMAP_REG_6 Register
Offset: 0x00050F7C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1132: EP_OUT_FLUSHED_SID_BITMAP_REG_7 Register
Offset: 0x00050F9C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1133: EP_OUT_FLUSHED_SID_BITMAP_REG_8 Register
Offset: 0x00050FBC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1134: EP_OUT_FLUSHED_SID_BITMAP_REG_9 Register
Offset: 0x00050FDC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1135: EP_OUT_FLUSHED_SID_BITMAP_REG_10 Register
Offset: 0x00050FFC

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1136: EP_OUT_FLUSHED_SID_BITMAP_REG_11 Register
Offset: 0x0005101C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1137: EP_OUT_FLUSHED_SID_BITMAP_REG_12 Register
Offset: 0x0005103C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Copyright © 2015 Marvell

Table 1138: EP_OUT_FLUSHED_SID_BITMAP_REG_13 Register
Offset: 0x0005105C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1139: EP_OUT_FLUSHED_SID_BITMAP_REG_14 Register
Offset: 0x0005107C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1140: EP_OUT_FLUSHED_SID_BITMAP_REG_15 Register
Offset: 0x0005109C

Bit	Field	Type/InitVal	Description
Internal use			
31:0	out_flushed_sid_bitmap_N	RO 0x0	out_flushed_sid_bitmap_N For debug purpose, indicate the bitmap of flushed SID.

Table 1141: EP_IN_NRDYQ_HEAD_POINTER_REG_1 Register
Offset: 0x000510BC

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP1 updated by HW.

Table 1142: EP_IN_NRDYQ_HEAD_POINTER_REG_2 Register
Offset: 0x000510C0

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP2 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1143: EP_IN_NRDYQ_HEAD_POINTER_REG_3 Register
Offset: 0x000510C4

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP3 updated by HW.

Table 1144: EP_IN_NRDYQ_HEAD_POINTER_REG_4 Register
Offset: 0x000510C8

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP4 updated by HW.

Table 1145: EP_IN_NRDYQ_HEAD_POINTER_REG_5 Register
Offset: 0x000510CC

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP5 updated by HW.

Table 1146: EP_IN_NRDYQ_HEAD_POINTER_REG_6 Register
Offset: 0x000510D0

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP6 updated by HW.

Table 1147: EP_IN_NRDYQ_HEAD_POINTER_REG_7 Register
Offset: 0x000510D4

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP7 updated by HW.

Copyright © 2015 Marvell

Table 1148: EP_IN_NRDYQ_HEAD_POINTER_REG_8 Register
Offset: 0x000510D8

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP8 updated by HW.

Table 1149: EP_IN_NRDYQ_HEAD_POINTER_REG_9 Register
Offset: 0x000510DC

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP9 updated by HW.

Table 1150: EP_IN_NRDYQ_HEAD_POINTER_REG_10 Register
Offset: 0x000510E0

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP10 updated by HW.

Table 1151: EP_IN_NRDYQ_HEAD_POINTER_REG_11 Register
Offset: 0x000510E4

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP11 updated by HW.

Table 1152: EP_IN_NRDYQ_HEAD_POINTER_REG_12 Register
Offset: 0x000510E8

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP12 updated by HW.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 1153: EP_IN_NRDYQ_HEAD_POINTER_REG_13 Register
Offset: 0x000510EC

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP13 updated by HW.

Table 1154: EP_IN_NRDYQ_HEAD_POINTER_REG_14 Register
Offset: 0x000510F0

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP14 updated by HW.

Table 1155: EP_IN_NRDYQ_HEAD_POINTER_REG_15 Register
Offset: 0x000510F4

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_head_ptr_N	RO 0x0	in_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding IN EP15 updated by HW.

Table 1156: EP_OUT_NRDYQ_HEAD_POINTER_REG_1 Register
Offset: 0x000510F8

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP1 updated by HW.

Table 1157: EP_OUT_NRDYQ_HEAD_POINTER_REG_2 Register
Offset: 0x000510FC

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP2 updated by HW.

Copyright © 2015 Marvell

Table 1158: EP_OUT_NRDYQ_HEAD_POINTER_REG_3 Register
Offset: 0x00051100

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP3 updated by HW.

Table 1159: EP_OUT_NRDYQ_HEAD_POINTER_REG_4 Register
Offset: 0x00051104

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP4 updated by HW.

Table 1160: EP_OUT_NRDYQ_HEAD_POINTER_REG_5 Register
Offset: 0x00051108

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP5 updated by HW.

Table 1161: EP_OUT_NRDYQ_HEAD_POINTER_REG_6 Register
Offset: 0x0005110C

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP6 updated by HW.

Table 1162: EP_OUT_NRDYQ_HEAD_POINTER_REG_7 Register
Offset: 0x00051110

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP7 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1163: EP_OUT_NRDYQ_HEAD_POINTER_REG_8 Register
Offset: 0x00051114

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP8 updated by HW.

Table 1164: EP_OUT_NRDYQ_HEAD_POINTER_REG_9 Register
Offset: 0x00051118

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP9 updated by HW.

Table 1165: EP_OUT_NRDYQ_HEAD_POINTER_REG_10 Register
Offset: 0x0005111C

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP10 updated by HW.

Table 1166: EP_OUT_NRDYQ_HEAD_POINTER_REG_11 Register
Offset: 0x00051120

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP11 updated by HW.

Table 1167: EP_OUT_NRDYQ_HEAD_POINTER_REG_12 Register
Offset: 0x00051124

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP12 updated by HW.

Copyright © 2015 Marvell

Table 1168: EP_OUT_NRDYQ_HEAD_POINTER_REG_13 Register
Offset: 0x00051128

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP13 updated by HW.

Table 1169: EP_OUT_NRDYQ_HEAD_POINTER_REG_14 Register
Offset: 0x0005112C

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP14 updated by HW.

Table 1170: EP_OUT_NRDYQ_HEAD_POINTER_REG_15 Register
Offset: 0x00051130

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_head_ptr_N	RO 0x0	out_nrdyq_head_ptr_N Used in bulk stream mode. NRDYQ head pointer of coorespoinding OUT EP15 updated by HW.

Table 1171: EP_IN_NRDYQ_TAIL_POINTER_REG_1 Register
Offset: 0x00051134

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP1 updated by HW.

Table 1172: EP_IN_NRDYQ_TAIL_POINTER_REG_2 Register
Offset: 0x00051138

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP2 updated by HW.

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1173: EP_IN_NRDYQ_TAIL_POINTER_REG_3 Register
Offset: 0x0005113C

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP3 updated by HW.

Table 1174: EP_IN_NRDYQ_TAIL_POINTER_REG_4 Register
Offset: 0x00051140

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP4 updated by HW.

Table 1175: EP_IN_NRDYQ_TAIL_POINTER_REG_5 Register
Offset: 0x00051144

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP5 updated by HW.

Table 1176: EP_IN_NRDYQ_TAIL_POINTER_REG_6 Register
Offset: 0x00051148

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP6 updated by HW.

Table 1177: EP_IN_NRDYQ_TAIL_POINTER_REG_7 Register
Offset: 0x0005114C

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP7 updated by HW.

Copyright © 2015 Marvell

Table 1178: EP_IN_NRDYQ_TAIL_POINTER_REG_8 Register
Offset: 0x00051150

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP8 updated by HW.

Table 1179: EP_IN_NRDYQ_TAIL_POINTER_REG_9 Register
Offset: 0x00051154

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP9 updated by HW.

Table 1180: EP_IN_NRDYQ_TAIL_POINTER_REG_10 Register
Offset: 0x00051158

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP10 updated by HW.

Table 1181: EP_IN_NRDYQ_TAIL_POINTER_REG_11 Register
Offset: 0x0005115C

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP11 updated by HW.

Table 1182: EP_IN_NRDYQ_TAIL_POINTER_REG_12 Register
Offset: 0x00051160

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP12 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1183: EP_IN_NRDYQ_TAIL_POINTER_REG_13 Register
Offset: 0x00051164

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP13 updated by HW.

Table 1184: EP_IN_NRDYQ_TAIL_POINTER_REG_14 Register
Offset: 0x00051168

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP14 updated by HW.

Table 1185: EP_IN_NRDYQ_TAIL_POINTER_REG_15 Register
Offset: 0x0005116C

Bit	Field	Type/InitVal	Description
31:0	in_nrdyq_tail_ptr_N	RO 0x0	in_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding IN EP15 updated by HW.

Table 1186: EP_OUT_NRDYQ_TAIL_POINTER_REG_1 Register
Offset: 0x00051170

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP1 updated by HW.

Table 1187: EP_OUT_NRDYQ_TAIL_POINTER_REG_2 Register
Offset: 0x00051174

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP2 updated by HW.

Copyright © 2015 Marvell

Table 1188: EP_OUT_NRDYQ_TAIL_POINTER_REG_3 Register
Offset: 0x00051178

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP3 updated by HW.

Table 1189: EP_OUT_NRDYQ_TAIL_POINTER_REG_4 Register
Offset: 0x0005117C

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP4 updated by HW.

Table 1190: EP_OUT_NRDYQ_TAIL_POINTER_REG_5 Register
Offset: 0x00051180

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP5 updated by HW.

Table 1191: EP_OUT_NRDYQ_TAIL_POINTER_REG_6 Register
Offset: 0x00051184

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP6 updated by HW.

Table 1192: EP_OUT_NRDYQ_TAIL_POINTER_REG_7 Register
Offset: 0x00051188

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP7 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1193: EP_OUT_NRDYQ_TAIL_POINTER_REG_8 Register
Offset: 0x0005118C

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP8 updated by HW.

Table 1194: EP_OUT_NRDYQ_TAIL_POINTER_REG_9 Register
Offset: 0x00051190

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP9 updated by HW.

Table 1195: EP_OUT_NRDYQ_TAIL_POINTER_REG_10 Register
Offset: 0x00051194

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP10 updated by HW.

Table 1196: EP_OUT_NRDYQ_TAIL_POINTER_REG_11 Register
Offset: 0x00051198

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP11 updated by HW.

Table 1197: EP_OUT_NRDYQ_TAIL_POINTER_REG_12 Register
Offset: 0x0005119C

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP12 updated by HW.

Copyright © 2015 Marvell

Table 1198: EP_OUT_NRDYQ_TAIL_POINTER_REG_13 Register
Offset: 0x000511A0

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP13 updated by HW.

Table 1199: EP_OUT_NRDYQ_TAIL_POINTER_REG_14 Register
Offset: 0x000511A4

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP14 updated by HW.

Table 1200: EP_OUT_NRDYQ_TAIL_POINTER_REG_15 Register
Offset: 0x000511A8

Bit	Field	Type/InitVal	Description
31:0	out_nrdyq_tail_ptr_N	RO 0x0	out_nrdyq_tail_ptr_N Used in bulk stream mode. NRDYQ tail pointer of cooresponding OUT EP15 updated by HW.

Table 1201: EP_IN_RDYQ_HEAD_POINTER_REG_1 Register
Offset: 0x000511AC

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP1 updated by HW.

Table 1202: EP_IN_RDYQ_HEAD_POINTER_REG_2 Register
Offset: 0x000511B0

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP2 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1203: EP_IN_RDYQ_HEAD_POINTER_REG_3 Register
Offset: 0x000511B4

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP3 updated by HW.

Table 1204: EP_IN_RDYQ_HEAD_POINTER_REG_4 Register
Offset: 0x000511B8

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP4 updated by HW.

Table 1205: EP_IN_RDYQ_HEAD_POINTER_REG_5 Register
Offset: 0x000511BC

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP1 updated by HW.

Table 1206: EP_IN_RDYQ_HEAD_POINTER_REG_6 Register
Offset: 0x000511C0

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP6 updated by HW.

Table 1207: EP_IN_RDYQ_HEAD_POINTER_REG_7 Register
Offset: 0x000511C4

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP7 updated by HW.

Copyright © 2015 Marvell

Table 1208: EP_IN_RDYQ_HEAD_POINTER_REG_8 Register
Offset: 0x000511C8

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP8 updated by HW.

Table 1209: EP_IN_RDYQ_HEAD_POINTER_REG_9 Register
Offset: 0x000511CC

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP9 updated by HW.

Table 1210: EP_IN_RDYQ_HEAD_POINTER_REG_10 Register
Offset: 0x000511D0

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP10 updated by HW.

Table 1211: EP_IN_RDYQ_HEAD_POINTER_REG_11 Register
Offset: 0x000511D4

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP11 updated by HW.

Table 1212: EP_IN_RDYQ_HEAD_POINTER_REG_12 Register
Offset: 0x000511D8

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP12 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1213: EP_IN_RDYQ_HEAD_POINTER_REG_13 Register
Offset: 0x000511DC

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP13 updated by HW.

Table 1214: EP_IN_RDYQ_HEAD_POINTER_REG_14 Register
Offset: 0x000511E0

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP14 updated by HW.

Table 1215: EP_IN_RDYQ_HEAD_POINTER_REG_15 Register
Offset: 0x000511E4

Bit	Field	Type/InitVal	Description
31:0	in_rdyq_head_ptr_N	RO 0x0	in_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding IN EP15 updated by HW.

Table 1216: EP_OUT_RDYQ_HEAD_POINTER_REG_1 Register
Offset: 0x000511E8

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP1 updated by HW.

Table 1217: EP_OUT_RDYQ_HEAD_POINTER_REG_2 Register
Offset: 0x000511EC

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP2 updated by HW.

Copyright © 2015 Marvell

Table 1218: EP_OUT_RDYQ_HEAD_POINTER_REG_3 Register
Offset: 0x000511F0

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP3 updated by HW.

Table 1219: EP_OUT_RDYQ_HEAD_POINTER_REG_4 Register
Offset: 0x000511F4

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP4 updated by HW.

Table 1220: EP_OUT_RDYQ_HEAD_POINTER_REG_5 Register
Offset: 0x000511F8

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP5 updated by HW.

Table 1221: EP_OUT_RDYQ_HEAD_POINTER_REG_6 Register
Offset: 0x000511FC

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP6 updated by HW.

Table 1222: EP_OUT_RDYQ_HEAD_POINTER_REG_7 Register
Offset: 0x00051200

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP7 updated by HW.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1223: EP_OUT_RDYQ_HEAD_POINTER_REG_8 Register
Offset: 0x00051204

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP8 updated by HW.

Table 1224: EP_OUT_RDYQ_HEAD_POINTER_REG_9 Register
Offset: 0x00051208

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP9 updated by HW.

Table 1225: EP_OUT_RDYQ_HEAD_POINTER_REG_10 Register
Offset: 0x0005120C

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP10 updated by HW.

Table 1226: EP_OUT_RDYQ_HEAD_POINTER_REG_11 Register
Offset: 0x00051210

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP11 updated by HW.

Table 1227: EP_OUT_RDYQ_HEAD_POINTER_REG_12 Register
Offset: 0x00051214

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP12 updated by HW.

Copyright © 2015 Marvell

Table 1228: EP_OUT_RDYQ_HEAD_POINTER_REG_13 Register
Offset: 0x00051218

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP13 updated by HW.

Table 1229: EP_OUT_RDYQ_HEAD_POINTER_REG_14 Register
Offset: 0x0005121C

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP14 updated by HW.

Table 1230: EP_OUT_RDYQ_HEAD_POINTER_REG_15 Register
Offset: 0x00051220

Bit	Field	Type/InitVal	Description
31:0	out_rdyq_head_ptr_N	RO 0x0	out_rdyq_head_ptr_N Used in bulk stream mode. RDYQ head pointer of cooresponding OUT EP15 updated by HW.

Table 1231: BULK_STREAMING_INT_STATUS_REG Register
Offset: 0x00051224

Bit	Field	Type/InitVal	Description
31:4	Reserved_1	RO 0x0	Reserved_1
3	prime_rcv	RO 0x0	prime_rcv Assert when there's any endpoint received prime.
2	himd_rcv	RO 0x0	himd_rcv Assert when there's any endpoint received himd.
1	stream_rej	RO 0x0	stream_rej Assert when any endpoint received stream reject.
0	host_flowctrl_pp0	RO 0x0	host_flowctrl_pp0 Assert when any endpoint received host flow control (pp=0) for stream id.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1232: BULK_STREAMING_INT_ENABLE_REG Register
Offset: 0x00051228

Bit	Field	Type/InitVal	Description
31:4	Reserved_1	RO 0x0	Reserved_1
3	prime_rcv_en	RW 0x0	prime_rcv_en interrupt enable for prime_rcv.
2	himd_rcv_en	RW 0x0	himd_rcv_en interrupt enable for himd_rcv.
1	stream_rej_en	RW 0x0	stream_rej_en interrupt enable for stream_rej.
0	host_flowctrl_pp0_en	RW 0x0	host_flowctrl_pp0_en interrupt enable for host_flowctrl_pp0.

Table 1233: PRIME_RCV_STATUS_REG Register
Offset: 0x0005122C

Bit	Field	Type/InitVal	Description
31	ep_in_prime_rcv_15	WO 0x0	ep_in_prime_rcv_15 Used in bulk stream mode. Indicating prime received for cooresponding endpoint.
30	ep_in_prime_rcv_14	WO 0x0	ep_in_prime_rcv_14
29	ep_in_prime_rcv_13	WO 0x0	ep_in_prime_rcv_13
28	ep_in_prime_rcv_12	WO 0x0	ep_in_prime_rcv_12
27	ep_in_prime_rcv_11	WO 0x0	ep_in_prime_rcv_11
26	ep_in_prime_rcv_10	WO 0x0	ep_in_prime_rcv_10
25	ep_in_prime_rcv_9	WO 0x0	ep_in_prime_rcv_9
24	ep_in_prime_rcv_8	WO 0x0	ep_in_prime_rcv_8
23	ep_in_prime_rcv_7	WO 0x0	ep_in_prime_rcv_7
22	ep_in_prime_rcv_6	WO 0x0	ep_in_prime_rcv_6
21	ep_in_prime_rcv_5	WO 0x0	ep_in_prime_rcv_5
20	ep_in_prime_rcv_4	WO 0x0	ep_in_prime_rcv_4

Copyright © 2015 Marvell

Table 1233: PRIME_RCV_STATUS_REG Register (Continued)
Offset: 0x0005122C

Bit	Field	Type/InitVal	Description
19	ep_in_prime_rcv_3	WO 0x0	ep_in_prime_rcv_3
18	ep_in_prime_rcv_2	WO 0x0	ep_in_prime_rcv_2
17	ep_in_prime_rcv_1	WO 0x0	ep_in_prime_rcv_1
16	Reserved_1	RO 0x0	Reserved_1
15	ep_out_prime_rcv_15	WO 0x0	ep_out_prime_rcv_15
14	ep_out_prime_rcv_14	WO 0x0	ep_out_prime_rcv_14
13	ep_out_prime_rcv_13	WO 0x0	ep_out_prime_rcv_13
12	ep_out_prime_rcv_12	WO 0x0	ep_out_prime_rcv_12
11	ep_out_prime_rcv_11	WO 0x0	ep_out_prime_rcv_11
10	ep_out_prime_rcv_10	WO 0x0	ep_out_prime_rcv_10
9	ep_out_prime_rcv_9	WO 0x0	ep_out_prime_rcv_9
8	ep_out_prime_rcv_8	WO 0x0	ep_out_prime_rcv_8
7	ep_out_prime_rcv_7	WO 0x0	ep_out_prime_rcv_7
6	ep_out_prime_rcv_6	WO 0x0	ep_out_prime_rcv_6
5	ep_out_prime_rcv_5	WO 0x0	ep_out_prime_rcv_5
4	ep_out_prime_rcv_4	WO 0x0	ep_out_prime_rcv_4
3	ep_out_prime_rcv_3	WO 0x0	ep_out_prime_rcv_3
2	ep_out_prime_rcv_2	WO 0x0	ep_out_prime_rcv_2
1	ep_out_prime_rcv_1	WO 0x0	ep_out_prime_rcv_1
0	Reserved_2	RO 0x0	Reserved_2

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1234: HIMD_RCV_STATUS_REG Register
Offset: 0x00051230

Bit	Field	Type/InitVal	Description
31	ep_in_himd_rcv_15	WO 0x0	ep_in_himd_rcv_15 Used in bulk stream mode. Indicating himd received for cooresponding endpoint.
30	ep_in_himd_rcv_14	WO 0x0	ep_in_himd_rcv_14
29	ep_in_himd_rcv_13	WO 0x0	ep_in_himd_rcv_13
28	ep_in_himd_rcv_12	WO 0x0	ep_in_himd_rcv_12
27	ep_in_himd_rcv_11	WO 0x0	ep_in_himd_rcv_11
26	ep_in_himd_rcv_10	WO 0x0	ep_in_himd_rcv_10
25	ep_in_himd_rcv_9	WO 0x0	ep_in_himd_rcv_9
24	ep_in_himd_rcv_8	WO 0x0	ep_in_himd_rcv_8
23	ep_in_himd_rcv_7	WO 0x0	ep_in_himd_rcv_7
22	ep_in_himd_rcv_6	WO 0x0	ep_in_himd_rcv_6
21	ep_in_himd_rcv_5	WO 0x0	ep_in_himd_rcv_5
20	ep_in_himd_rcv_4	WO 0x0	ep_in_himd_rcv_4
19	ep_in_himd_rcv_3	WO 0x0	ep_in_himd_rcv_3
18	ep_in_himd_rcv_2	WO 0x0	ep_in_himd_rcv_2
17	ep_in_himd_rcv_1	WO 0x0	ep_in_himd_rcv_1
16	Reserved_1	WO 0x0	Reserved_1
15	ep_out_himd_rcv_15	WO 0x0	ep_out_himd_rcv_15
14	ep_out_himd_rcv_14	WO 0x0	ep_out_himd_rcv_14
13	ep_out_himd_rcv_13	WO 0x0	ep_out_himd_rcv_13
12	ep_out_himd_rcv_12	WO 0x0	ep_out_himd_rcv_12

Copyright © 2015 Marvell

Table 1234: HIMD_RCV_STATUS_REG Register (Continued)
Offset: 0x00051230

Bit	Field	Type/InitVal	Description
11	ep_out_himd_rcv_11	WO 0x0	ep_out_himd_rcv_11
10	ep_out_himd_rcv_10	WO 0x0	ep_out_himd_rcv_10
9	ep_out_himd_rcv_9	WO 0x0	ep_out_himd_rcv_9
8	ep_out_himd_rcv_8	WO 0x0	ep_out_himd_rcv_8
7	ep_out_himd_rcv_7	WO 0x0	ep_out_himd_rcv_7
6	ep_out_himd_rcv_6	WO 0x0	ep_out_himd_rcv_6
5	ep_out_himd_rcv_5	WO 0x0	ep_out_himd_rcv_5
4	ep_out_himd_rcv_4	WO 0x0	ep_out_himd_rcv_4
3	ep_out_himd_rcv_3	WO 0x0	ep_out_himd_rcv_3
2	ep_out_himd_rcv_2	WO 0x0	ep_out_himd_rcv_2
1	ep_out_himd_rcv_1	WO 0x0	ep_out_himd_rcv_1
0	Reserved_2	RO 0x0	Reserved_2

Table 1235: STREAM_REJ_STATUS_REG Register
Offset: 0x00051234

Bit	Field	Type/InitVal	Description
31	ep_in_stream_rej_15	WO 0x0	ep_in_stream_rej_15 Used in bulk stream mode. Indicating stream is rejected for cooresponding endpoint.
30	ep_in_stream_rej_14	WO 0x0	ep_in_stream_rej_14
29	ep_in_stream_rej_13	WO 0x0	ep_in_stream_rej_13
28	ep_in_stream_rej_12	WO 0x0	ep_in_stream_rej_12
27	ep_in_stream_rej_11	WO 0x0	ep_in_stream_rej_11

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1235: STREAM_REJ_STATUS_REG Register (Continued)
Offset: 0x00051234

Bit	Field	Type/InitVal	Description
26	ep_in_stream_rej_10	WO 0x0	ep_in_stream_rej_10
25	ep_in_stream_rej_9	WO 0x0	ep_in_stream_rej_9
24	ep_in_stream_rej_8	WO 0x0	ep_in_stream_rej_8
23	ep_in_stream_rej_7	WO 0x0	ep_in_stream_rej_7
22	ep_in_stream_rej_6	WO 0x0	ep_in_stream_rej_6
21	ep_in_stream_rej_5	WO 0x0	ep_in_stream_rej_5
20	ep_in_stream_rej_4	WO 0x0	ep_in_stream_rej_4
19	ep_in_stream_rej_3	WO 0x0	ep_in_stream_rej_3
18	ep_in_stream_rej_2	WO 0x0	ep_in_stream_rej_2
17	ep_in_stream_rej_1	WO 0x0	ep_in_stream_rej_1
16	Reserved_1	RO 0x0	Reserved_1
15	ep_out_stream_rej_15	WO 0x0	ep_out_stream_rej_15
14	ep_out_stream_rej_14	WO 0x0	ep_out_stream_rej_14
13	ep_out_stream_rej_13	WO 0x0	ep_out_stream_rej_13
12	ep_out_stream_rej_12	WO 0x0	ep_out_stream_rej_12
11	ep_out_stream_rej_11	WO 0x0	ep_out_stream_rej_11
10	ep_out_stream_rej_10	WO 0x0	ep_out_stream_rej_10
9	ep_out_stream_rej_9	WO 0x0	ep_out_stream_rej_9
8	ep_out_stream_rej_8	WO 0x0	ep_out_stream_rej_8
7	ep_out_stream_rej_7	WO 0x0	ep_out_stream_rej_7

Copyright © 2015 Marvell

Table 1235: STREAM_REJ_STATUS_REG Register (Continued)
Offset: 0x00051234

Bit	Field	Type/InitVal	Description
6	ep_out_stream_rej_6	WO 0x0	ep_out_stream_rej_6
5	ep_out_stream_rej_5	WO 0x0	ep_out_stream_rej_5
4	ep_out_stream_rej_4	WO 0x0	ep_out_stream_rej_4
3	ep_out_stream_rej_3	WO 0x0	ep_out_stream_rej_3
2	ep_out_stream_rej_2	WO 0x0	ep_out_stream_rej_2
1	ep_out_stream_rej_1	WO 0x0	ep_out_stream_rej_1
0	Reserved_2	RO 0x0	Reserved_2

Table 1236: HOST_FLOWCTRL_STATUS_REG Register
Offset: 0x00051238

Bit	Field	Type/InitVal	Description
31	ep_in_host_flowctlr_15	WO 0x0	ep_in_host_flowctlr_15 Used in bulk stream mode. Indicating host flow control current stream id for cooresponding endpoint.
30	ep_in_host_flowctlr_14	WO 0x0	ep_in_host_flowctlr_14
29	ep_in_host_flowctlr_13	WO 0x0	ep_in_host_flowctlr_13
28	ep_in_host_flowctlr_12	WO 0x0	ep_in_host_flowctlr_12
27	ep_in_host_flowctlr_11	WO 0x0	ep_in_host_flowctlr_11
26	ep_in_host_flowctlr_10	WO 0x0	ep_in_host_flowctlr_10
25	ep_in_host_flowctlr_9	WO 0x0	ep_in_host_flowctlr_9
24	ep_in_host_flowctlr_8	WO 0x0	ep_in_host_flowctlr_8
23	ep_in_host_flowctlr_7	WO 0x0	ep_in_host_flowctlr_7
22	ep_in_host_flowctlr_6	WO 0x0	ep_in_host_flowctlr_6

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1236: HOST_FLOWCTRL_STATUS_REG Register (Continued)
Offset: 0x00051238

Bit	Field	Type/InitVal	Description
21	ep_in_host_flowctlr_5	WO 0x0	ep_in_host_flowctlr_5
20	ep_in_host_flowctlr_4	WO 0x0	ep_in_host_flowctlr_4
19	ep_in_host_flowctlr_3	WO 0x0	ep_in_host_flowctlr_3
18	ep_in_host_flowctlr_2	WO 0x0	ep_in_host_flowctlr_2
17	ep_in_host_flowctlr_1	WO 0x0	ep_in_host_flowctlr_1
16	Reserved_1	RO 0x0	Reserved_1
15	ep_out_host_flowctlr_15	WO 0x0	ep_out_host_flowctlr_15
14	ep_out_host_flowctlr_14	WO 0x0	ep_out_host_flowctlr_14
13	ep_out_host_flowctlr_13	WO 0x0	ep_out_host_flowctlr_13
12	ep_out_host_flowctlr_12	WO 0x0	ep_out_host_flowctlr_12
11	ep_out_host_flowctlr_11	WO 0x0	ep_out_host_flowctlr_11
10	ep_out_host_flowctlr_10	WO 0x0	ep_out_host_flowctlr_10
9	ep_out_host_flowctlr_9	WO 0x0	ep_out_host_flowctlr_9
8	ep_out_host_flowctlr_8	WO 0x0	ep_out_host_flowctlr_8
7	ep_out_host_flowctlr_7	WO 0x0	ep_out_host_flowctlr_7
6	ep_out_host_flowctlr_6	WO 0x0	ep_out_host_flowctlr_6
5	ep_out_host_flowctlr_5	WO 0x0	ep_out_host_flowctlr_5
4	ep_out_host_flowctlr_4	WO 0x0	ep_out_host_flowctlr_4
3	ep_out_host_flowctlr_3	WO 0x0	ep_out_host_flowctlr_3
2	ep_out_host_flowctlr_2	WO 0x0	ep_out_host_flowctlr_2

Copyright © 2015 Marvell

Table 1236: HOST_FLOWCTRL_STATUS_REG Register (Continued)
Offset: 0x00051238

Bit	Field	Type/InitVal	Description
1	ep_out_host_flowctlr_1	WO 0x0	ep_out_host_flowctlr_1
0	Reserved_2	RO 0x0	Reserved_2

Table 1237: BD_MEM_IN_REG_1 Register
Offset: 0x0005123C

Bit	Field	Type/InitVal	Description
Debug register			
31:16	dbg_dma_in_cache_rdata	RO 0x0	dbg_dma_in_cache_rdata This field shows the lower 16 bits of BD address of selected endpoint (selected by dbg_dma_mux) and cache index (dbg_in_cache_ptr).
15:0	dbg_dma_out_cache_rdata	RO 0x0	dbg_dma_out_cache_rdata This field shows the lower 16 bits of BD address of selected endpoint (selected by dbg_dma_mux) and cache index (dbg_out_cache_ptr).

Table 1238: PL_DEBUG_REG_1 Register
Offset: 0x00051264

Bit	Field	Type/InitVal	Description
Debug register			
31:0	debug_for_in_ep_31_0	RO 0x0	debug_for_in_ep[31:0] debug signals.

Table 1239: PL_DEBUG_REG_2 Register
Offset: 0x00051268

Bit	Field	Type/InitVal	Description
Debug register			
31:29	Reserved_1	RO 0x0	Reserved_1 debug signals
28:0	debug_for_in_ep_60_32	RO 0x0	debug_for_in_ep[60:32]

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell

Table 1240: PL_DEBUG_REG_3 Register
Offset: 0x0005126C

Bit	Field	Type/InitVal	Description
Debug register			
31:0	debug_for_out_ep_31_0	RO 0x0	debug_for_out_ep[31:0] debug signals

Table 1241: PL_DEBUG_REG_4 Register
Offset: 0x00051270

Bit	Field	Type/InitVal	Description
Debug register			
31:13	Reserved_1	RO 0x0	Reserved_1 debug signals
12:0	debug_for_out_ep_44_32	RO 0x0	debug_for_out_ep[44:32]

Table 1242: DMA_DBG_IN_REG_1 Register
Offset: 0x00051274

Bit	Field	Type/InitVal	Description
Debug register			
31:0	dbg_st_dma_in_31_0	RO 0x0	dbg_st_dma_in[31:0] debug signals

Table 1243: DMA_DBG_IN_REG_2 Register
Offset: 0x00051278

Bit	Field	Type/InitVal	Description
Debug register			
31:16	Reserved	RO 0x0	Reserved
15:0	dbg_st_dma_in_47_32	RO 0x0	dbg_st_dma_in[47:32] debug signals

Copyright © 2015 Marvell

Table 1244: DMA_DBG_OUT_REG_1 Register
Offset: 0x0005127C

Bit	Field	Type/InitVal	Description
Debug register			
31:0	dbg_st_dma_out_31_0	RO 0x0	dbg_st_dma_out[31:0] debug signals

Table 1245: DMA_DBG_OUT_REG_2 Register
Offset: 0x00051280

Bit	Field	Type/InitVal	Description
Debug register			
31:16	Reserved	RO 0x0	Reserved
15:0	dbg_st_dma_out_47_32	RO 0x0	dbg_st_dma_out[47:32] debug signals

A.13.1.7 USB 2 Modified

Table 1246: USB2_USBSTS Register
Offset: 0x00054144

Bit	Field	Type/InitVal	Description
31:26	reserved3	RO 0x0	reserved3
25	TI1	RW 0x0	Timer Interrupt 1 Refer to ChipIdea (Atlantic core) Programmer's Manual
24	TI0	RW 0x0	Timer Interrupt 0 Refer to ChipIdea (Atlantic core) Programmer's Manual
23:20	reserved2	RO 0x0	reserved 2
19	UPI	RO 0x0	UPI Refer to ChipIdea (Atlantic core) Programmer's Manual
18	UAI	RO 0x0	UAI Refer to ChipIdea (Atlantic core) Programmer's Manual
17	reserved1	RO 0x0	reserved1
16	NAKI	RO 0x0	NAK Interrupt Refer to ChipIdea (Atlantic core) Programmer's Manual
15	AS	RO 0x0	AS Refer to ChipIdea (Atlantic core) Programmer's Manual
14	PS	RO 0x0	PS Refer to ChipIdea (Atlantic core) Programmer's Manual

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1246: USB2_USBSTS Register (Continued)
Offset: 0x00054144

Bit	Field	Type/InitVal	Description
13	RCL	RO 0x0	RCL Refer to ChipIdea (Atlantic core) Programmer's Manual
12	HCH	RO 0x0	HCH Refer to ChipIdea (Atlantic core) Programmer's Manual
11	UALTI	RO 0x0	UALTI Refer to ChipIdea (Atlantic core) Programmer's Manual
10	ULPII	RO 0x0	ULPII Refer to ChipIdea (Atlantic core) Programmer's Manual
9	LI_entry_Interrupt	RW 0x0	L1_entry_Interrupt L1 entry Interrupt
8	SLI	RW 0x0	DCSuspend Refer to ChipIdea (Atlantic core) Programmer's Manual
7	SRI	RW 0x0	SOF Received Refer to ChipIdea (Atlantic core) Programmer's Manual
6	URI	RW 0x0	USB Reset Received Refer to ChipIdea (Atlantic core) Programmer's Manual
5	AAI	RO 0x0	AAI Refer to ChipIdea (Atlantic core) Programmer's Manual
4	SEI	RW 0x0	System Error Refer to ChipIdea (Atlantic core) Programmer's Manual
3	FRI	RO 0x0	FRI Refer to ChipIdea (Atlantic core) Programmer's Manual
2	PCI	RW 0x0	Port Change Detect Refer to Chipidea(Atlantic core) Programmer's manual. Also L1_exit will trigger this bit to 1.
1	UEI	RW 0x0	USB Error Interrupt Refer to ChipIdea (Atlantic core) Programmer's Manual
0	UI	RW 0x0	USB Interrupt Refer to ChipIdea (Atlantic core) Programmer's Manual

Table 1247: USB2_USBINT Register
Offset: 0x00054148

Bit	Field	Type/InitVal	Description
31:26	reserved4	RO 0x0	reserved4
25	TIE1	RW 0x0	Timer Interrupt Enable 1 Refer to ChipIdea (Atlantic core) Programmer's Manual
24	TIE0	RW 0x0	Timer Interrupt Enable 0 Refer to ChipIdea (Atlantic core) Programmer's Manual
23:20	reserved3	RO 0x0	reserved3

Copyright © 2015 Marvell

Table 1247: USB2_USBINT Register (Continued)
Offset: 0x00054148

Bit	Field	Type/InitVal	Description
19	UPIE	RW 0x0	Host Periodic Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
18	UAIE	RW 0x0	Host Async Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
17	reserved2	RO 0x0	reserved2
16	NAKE	RW 0x0	NAK Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
15:10	reserved1	RO 0x0	reserved1
9	L1E	RW 0x0	L1_entry Interrupt Enable Setting this bit to 1 enables L1_entry interrupt.
8	SLE	RW 0x0	DCSuspend Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
7	SRE	RW 0x0	SOF Received Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
6	URE	RW 0x0	Reset Received Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
5	AAE	RO 0x0	AAE Refer to ChipIdea (Atlantic core) Programmer's Manual
4	SEE	RW 0x0	System Error Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
3	FRE	RO 0x0	FRE Refer to ChipIdea (Atlantic core) Programmer's Manual
2	PCE	RW 0x0	Port Change Detect Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
1	UEE	RW 0x0	USB Error Interrupt Refer to ChipIdea (Atlantic core) Programmer's Manual
0	UE	RW 0x0	USB Interrupt Enable Refer to ChipIdea (Atlantic core) Programmer's Manual

Table 1248: USB2_DEVLC Register
Offset: 0x0005415C

Bit	Field	Type/InitVal	Description
This register is for USB2 LPM control.			
31:19	reserved1	RO 0x0	reserved1
18	LPM_EN	RW 0x1	LPM_EN Set to 1 to enable LPM feature.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary
Not Approved by Document Control. For Review Only.

Table 1248: USB2_DEVLC Register (Continued)
Offset: 0x0005415C

Bit	Field	Type/InitVal	Description
17	reserved2	RO 0x0	reserved2
16	STL	RW 0x0	STL STALL reply. When this bit is set, device controller will stall all incoming LPM packets.
15:12	reserved3	RO 0x0	reserved3
11:1	BA	RO 0x0	BA This is the bmAttributes field of the LPM sub-token received, after the EXT PID token.
0	NYT	RW 0x0	NYT NYET reply. When this bit is set, and the condition to send STALL is not met, and bit 16 field STL is not set, device controller will NYET all the LPM tokens.

Table 1249: USB2_PORTSC Register
Offset: 0x00054184

Bit	Field	Type/InitVal	Description
31:30	PTS	RW 0x0	Parallel Transceiver Select Refer to ChipIdea (Atlantic core) Programmer's Manual
29	STS	RO 0x0	Serial Transceiver Select Refer to ChipIdea (Atlantic core) Programmer's Manual
28	PTW	RO 0x0	Parallel Transceiver Width Refer to ChipIdea (Atlantic core) Programmer's Manual
27:26	PSPD	RO 0x3	Port Speed Refer to ChipIdea (Atlantic core) Programmer's Manual
25	PTS2	RO 0x0	Parallel Transceiver Select Refer to ChipIdea (Atlantic core) Programmer's Manual
24	PFSC	RW 0x0	Port Force Full Speed Connect Refer to ChipIdea (Atlantic core) Programmer's Manual
23	PHCD	RW 0x0	Port Low Power Clock Disable Refer to ChipIdea (Atlantic core) Programmer's Manual
22	WKOC	RW 0x0	Wake on Over-current Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
21	WKDS	RW 0x0	Wake on Disconnect Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
20	WKCN	RW 0x0	Wake on Connect Enable Refer to ChipIdea (Atlantic core) Programmer's Manual
19:16	PTC	RW 0x0	Port Test Control Refer to ChipIdea (Atlantic core) Programmer's Manual
15:14	PIC	RW 0x0	Port Indicator Control Refer to ChipIdea (Atlantic core) Programmer's Manual

Copyright © 2015 Marvell

Table 1249: USB2_PORTSC Register (Continued)
Offset: 0x00054184

Bit	Field	Type/InitVal	Description
13	reserved	RO 0x0	reserved Refer to ChipIdea (Atlantic core) Programmer's Manual
12	L1_STATE	RO 0x0	L1_STATE Value "1" indicates device controller is in L1 state.
11:10	LS	RO 0x0	Line Status Refer to ChipIdea (Atlantic core) Programmer's Manual
9	HSP	RO 0x0	High-Speed Port Refer to ChipIdea (Atlantic core) Programmer's Manual
8	PR	RW 0x0	Port Reset Refer to ChipIdea (Atlantic core) Programmer's Manual
7	SUSP	RW 0x0	Suspend Refer to ChipIdea (Atlantic core) Programmer's Manual
6	FPR	RW 0x0	Force Port Resume Refer to ChipIdea (Atlantic core) Programmer's Manual
5	OCC	RO 0x0	Over-current Change Refer to ChipIdea (Atlantic core) Programmer's Manual
4	OCA	RO 0x0	Over-current Active Refer to ChipIdea (Atlantic core) Programmer's Manual
3	PEC	RO 0x0	Port Enabled Change Refer to ChipIdea (Atlantic core) Programmer's Manual
2	PE	RO 0x1	Port Enabled Refer to ChipIdea (Atlantic core) Programmer's Manual
1	reserved1	RO 0x0	reserved1
0	CCS	RO 0x0	Current Connect Status Refer to ChipIdea (Atlantic core) Programmer's Manual

Document Classification: Proprietary Information
Not Approved by Document Control. For Review Only.
Copyright © 2015 Marvell
November 25, 2015, Preliminary

A.14 SATA 3.0 Host Controller Registers

The following table provides a summarized list of all registers that belong to the SATA 3.0 Host Controller, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1250: Summary Map Table for the SATA 3.0 Host Controller Registers

Register Name	Offset	Table and Page
SATA 3.0 Host Controller		
SATA3 General		
SoC Capability Register (i=0–1)	Unit0: 0x000A8000, Unit1: 0x000E0000	Table 1251, p. 1029
Global SoC Control Register (i=0–1)	Unit0: 0x000A8004, Unit1: 0x000E0004	Table 1252, p. 1030
Interrupt Status Register (i=0–1)	Unit0: 0x000A8008, Unit1: 0x000E0008	Table 1253, p. 1030
Ports Implemented Register (i=0–1)	Unit0: 0x000A800C, Unit1: 0x000E000C	Table 1254, p. 1031
AHCI Version Register (i=0–1)	Unit0: 0x000A8010, Unit1: 0x000E0010	Table 1255, p. 1031
Command Completion Coalescing Control Register (i=0–1)	Unit0: 0x000A8014, Unit1: 0x000E0014	Table 1256, p. 1031
Command Completion Coalescing Ports Register (i=0–1)	Unit0: 0x000A8018, Unit1: 0x000E0018	Table 1257, p. 1032
Vendor-Specific 0 Address Register (i=0–1)	Unit0: 0x000A80A0, Unit1: 0x000E00A0	Table 1258, p. 1033
Vendor-Specific 0 Data Register (i=0–1)	Unit0: 0x000A80A4, Unit1: 0x000E00A4	Table 1259, p. 1033
SATA3 Arbiter		
Window0 Control Register (i=0–1)	Unit0: 0x000A8060, Unit1: 0x000E0060	Table 1260, p. 1033
Window0 Base Register (i=0–1)	Unit0: 0x000A8064, Unit1: 0x000E0064	Table 1261, p. 1034
Window0 Size Register (i=0–1)	Unit0: 0x000A8068, Unit1: 0x000E0068	Table 1262, p. 1034
Window1 Control Register (i=0–1)	Unit0: 0x000A8070, Unit1: 0x000E0070	Table 1263, p. 1035
Window1 Base Register (i=0–1)	Unit0: 0x000A8074, Unit1: 0x000E0074	Table 1264, p. 1035
Window1 Size Register (i=0–1)	Unit0: 0x000A8078, Unit1: 0x000E0078	Table 1265, p. 1035
Window2 Control Register (i=0–1)	Unit0: 0x000A8080, Unit1: 0x000E0080	Table 1266, p. 1035
Window2 Base Register (i=0–1)	Unit0: 0x000A8084, Unit1: 0x000E0084	Table 1267, p. 1036
Window2 Size Register (i=0–1)	Unit0: 0x000A8088, Unit1: 0x000E0088	Table 1268, p. 1036

Copyright © 2015 Marvell

Table 1250: Summary Map Table for the SATA 3.0 Host Controller Registers (Continued)

Register Name	Offset	Table and Page
Window3 Control Register (i=0–1)	Unit0: 0x000A8090, Unit1: 0x000E0090	Table 1269, p. 1036
Window3 Base Register (i=0–1)	Unit0: 0x000A8094, Unit1: 0x000E0094	Table 1270, p. 1037
Window3 Size Register (i=0–1)	Unit0: 0x000A8098, Unit1: 0x000E0098	Table 1271, p. 1037
SATA3 Port		
Command List Base Address Lower 32-bits Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8100, Unit0 Port1: 0x000A8180, Unit1 Port0: 0x000E0100, Unit1 Port1: 0x000E0180	Table 1272, p. 1037
SATA control register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8100, Unit0 Port1: 0x000A8180, Unit1 Port0: 0x000E0100, Unit1 Port1: 0x000E0180	Table 1273, p. 1038
MBUS request size and interface select register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8104, Unit0 Port1: 0x000A8184, Unit1 Port0: 0x000E0104, Unit1 Port1: 0x000E0184	Table 1274, p. 1039
Command List Base Address Upper 32-bits Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8104, Unit0 Port1: 0x000A8184, Unit1 Port0: 0x000E0104, Unit1 Port1: 0x000E0184	Table 1275, p. 1039
SATA memory timing control register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8108, Unit0 Port1: 0x000A8188, Unit1 Port0: 0x000E0108, Unit1 Port1: 0x000E0188	Table 1276, p. 1039
Received FIS Base Address Lower 32-bits Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8108, Unit0 Port1: 0x000A8188, Unit1 Port0: 0x000E0108, Unit1 Port1: 0x000E0188	Table 1277, p. 1040
Received FIS Base Address Upper 32-bits Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A810C, Unit0 Port1: 0x000A818C, Unit1 Port0: 0x000E010C, Unit1 Port1: 0x000E018C	Table 1278, p. 1040
Port Interrupt Status Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8110, Unit0 Port1: 0x000A8190, Unit1 Port0: 0x000E0110, Unit1 Port1: 0x000E0190	Table 1279, p. 1040
Port Interrupt Enable Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8114, Unit0 Port1: 0x000A8194, Unit1 Port0: 0x000E0114, Unit1 Port1: 0x000E0194	Table 1280, p. 1043
Port Command and Status Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8118, Unit0 Port1: 0x000A8198, Unit1 Port0: 0x000E0118, Unit1 Port1: 0x000E0198	Table 1281, p. 1045
Reserved 1 Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A811C, Unit0 Port1: 0x000A819C, Unit1 Port0: 0x000E011C, Unit1 Port1: 0x000E019C	Table 1282, p. 1047

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1250: Summary Map Table for the SATA 3.0 Host Controller Registers (Continued)

Register Name	Offset	Table and Page
Port Task File Data Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8120, Unit0 Port1: 0x000A81A0, Unit1 Port0: 0x000E0120, Unit1 Port1: 0x000E01A0	Table 1283, p. 1047
Port Signature Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8124, Unit0 Port1: 0x000A81A4, Unit1 Port0: 0x000E0124, Unit1 Port1: 0x000E01A4	Table 1284, p. 1048
SATA Status (SStatus/SCR0) Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8128, Unit0 Port1: 0x000A81A8, Unit1 Port0: 0x000E0128, Unit1 Port1: 0x000E01A8	Table 1285, p. 1048
SATA Control (SControl/SCR2) Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A812C, Unit0 Port1: 0x000A81AC, Unit1 Port0: 0x000E012C, Unit1 Port1: 0x000E01AC	Table 1286, p. 1049
SATA Error (SError/SCR1) Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8130, Unit0 Port1: 0x000A81B0, Unit1 Port0: 0x000E0130, Unit1 Port1: 0x000E01B0	Table 1287, p. 1050
SATA Active (SActive/SCR3) Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8134, Unit0 Port1: 0x000A81B4, Unit1 Port0: 0x000E0134, Unit1 Port1: 0x000E01B4	Table 1288, p. 1052
Command Issue Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8138, Unit0 Port1: 0x000A81B8, Unit1 Port0: 0x000E0138, Unit1 Port1: 0x000E01B8	Table 1289, p. 1052
Port SATA Notification (SNotification/SCR4) Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A813C, Unit0 Port1: 0x000A81BC, Unit1 Port0: 0x000E013C, Unit1 Port1: 0x000E01BC	Table 1290, p. 1052
Port FIS-Based Switching Control Register (i=0–1, n=0–1)	Unit0 Port0: 0x000A8140, Unit0 Port1: 0x000A81C0, Unit1 Port0: 0x000E0140, Unit1 Port1: 0x000E01C0	Table 1291, p. 1053

A.14.1 SATA 3.0 Host Controller

A.14.1.1 SATA3 General

Table 1251: SoC Capability Register (i=0–1)
Offset: Unit0: 0x000A8000, Unit1: 0x000E0000
Offset Formula: 0x00000000+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:19	Reserved	RSVD 0x1C46	Reserved

Table 1251: SoC Capability Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8000, Unit1: 0x000E0000
Offset Formula: 0x00000000+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
18	AHCI Only	RO 0x1	Enable support for AHCI Only. 0 = Disable: Supports legacy interface in addition to AHCI. 1 = Enable: Supports only AHCI.
17:5	Reserved	RO 0x1FF9	Reserved
4:0	Ports	RO 0x1	Ports. The number of ports supported by the controller using the Enhanced mode registers. The register values are zero-based. The other values are reserved. 0 = Supports 1 SATA port 1 = Supports 2 SATA ports 3 = Supports 4 SATA ports

Table 1252: Global SoC Control Register (i=0–1)
Offset: Unit0: 0x000A8004, Unit1: 0x000E0004
Offset Formula: 0x00000004+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31	AHCI En	RW 0x0	AHCI Enable. 0 = Disable: AHCI disabled. The software must use legacy mechanisms to communicate with the SoC. 1 = Enable: AHCI enabled.
30:3	Reserved	RO 0x0	Reserved
2	MSI RSM	RO 0x0	Message Signaled Interrupt Revert to Single Message Mode. 0 = Disable: The SoC has not reverted to single MSI mode. 1 = Enable: Single MSI mode is in use.
1	Int En	RW 0x0	Interrupt Enable. 0 = Disable: Disable interrupts from all ports. 1 = Enable: Enable interrupts.
0	SoC ResetT	RW 0x0	SoC Reset. When set to 1 by software, this bit causes an internal reset of the SoC. When the SoC has performed the reset action, this bit is cleared to 0. A software write of 0 has no effect.

Table 1253: Interrupt Status Register (i=0–1)
Offset: Unit0: 0x000A8008, Unit1: 0x000E0008
Offset Formula: 0x00000008+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:3	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 1253: Interrupt Status Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8008, Unit1: 0x000E0008
Offset Formula: 0x00000008+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
2	Reserved	RSVD 0x0	Reserved
1	Port 1 Int Status	RW1C 0x0	Port 1 Interrupt Pending Status.
0	Port 0 Int Status	RW1C 0x0	Port 0 Interrupt Pending Status.

Table 1254: Ports Implemented Register (i=0–1)
Offset: Unit0: 0x000A800C, Unit1: 0x000E000C
Offset Formula: 0x0000000C+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:2	Reserved	RO 0x0	Reserved
1:0	Ports Implemented	RW 0x3	Ports Implemented. Indicates which ports are implemented. Bit [0] corresponds to port 0. Bit [1] corresponds to port 1. 1 = One port: A single port is supported. 3 = Two ports: Two ports are supported.

Table 1255: AHCI Version Register (i=0–1)
Offset: Unit0: 0x000A8010, Unit1: 0x000E0010
Offset Formula: 0x00000010+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:0	AHCI Version	RO 0x00010000	AHCI Version.

Table 1256: Command Completion Coalescing Control Register (i=0–1)
Offset: Unit0: 0x000A8014, Unit1: 0x000E0014
Offset Formula: 0x00000014+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	CCC Timeout	RW 0x1	The timeout value is specified in 1 millisecond intervals. The timer accuracy shall be within 5%. hCccTimer is loaded with this timeout value. hCccTimer is only decremented when commands are outstanding on selected ports. The SoC will signal a CCC interrupt when hCccTimer has decremented to 0. hCccTimer is reset to the timeout value on the assertion of each CCC interrupt. A timeout value of 0 is reserved.

Copyright © 2015 Marvell

Table 1256: Command Completion Coalescing Control Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8014, Unit1: 0x000E0014
Offset Formula: 0x00000014+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
15:8	Cmd Comp	RW 0x1	Specifies the number of command completions that are necessary to cause a Command Coalescing Control (CCC) interrupt. The SoC has an internal command completion counter, hCccComplete. hCccComplete is incremented by 1 each time a selected port has a command completion. When hCccComplete is equal to the command completions value, a CCC interrupt is signaled. The internal command completion counter is reset to 0 on the assertion of each CCC interrupt. A value of 0 for this field disables CCC interrupt generation based on the number of commands completed, that is, CCC interrupts are only generated based on the timer.
7:3	CCC Interrupt	RO 0x4	Specifies the interrupt used by the CCC feature. This interrupt must be marked as unused in the Ports Implemented register by the corresponding bit being set to 0. Thus, the CCC interrupt corresponds to the interrupt for an unimplemented port on the controller. When a CCC interrupt occurs, the IS.IPS[INT] bit is asserted. This field also specifies the interrupt vector used for MSI.
2:1	Reserved	RO 0x0	Reserved
0	CCC Enable	RO 0x0	Enable When cleared to, the command completion coalescing feature is disabled and no CCC interrupts are generated. When set to 1, the command completion coalescing feature is enabled and CCC interrupts may be generated based on timeout or command completion conditions. Software only changes the contents of the TV and CC fields when EN is cleared to 0. On transition of this bit from 0 to 1, any updated values for the TV and CC fields are implemented. 0 = Disable 1 = Enable

Table 1257: Command Completion Coalescing Ports Register (i=0–1)
Offset: Unit0: 0x000A8018, Unit1: 0x000E0018
Offset Formula: 0x00000018+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:2	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 1257: Command Completion Coalescing Ports Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8018, Unit1: 0x000E0018
Offset Formula: 0x00000018+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
1	Port1 CCC Enable	RW 0x0	If a field is set to 1, port 1 is part of the command completion coalescing feature. If a field is cleared to 0, port 1 is not part of the command completion coalescing feature. Bits set to 1 in this register must also have the corresponding bit set to 1 in the Ports Implemented register. An updated value for this field takes effect within one timer increment (1 ms). 0 = Disable 1 = Enable
0	Port0 CCC Enable	RW 0x0	If a field is set to 1, the port 0 is part of the command completion coalescing feature. If a field is cleared to 0, the port 0 is not part of the command completion coalescing feature. Bits set to 1 in this register must also have the corresponding bit set to 1 in the Ports Implemented register. An updated value for this field takes effect within one timer increment (1 ms). 0 = Disable 1 = Enable

Table 1258: Vendor-Specific 0 Address Register (i=0–1)
Offset: Unit0: 0x000A80A0, Unit1: 0x000E00A0
Offset Formula: 0x000000A0+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:0	Vend Sp Addr 0	RW 0x0	Vendor Specific Address 0. Vend Sp Addr 0 specifies the address from 00h to 80h.

Table 1259: Vendor-Specific 0 Data Register (i=0–1)
Offset: Unit0: 0x000A80A4, Unit1: 0x000E00A4
Offset Formula: 0x000000A4+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:0	Vend Sp Data 0	RW 0x0	Vendor Specific Data 0.

A.14.1.2 SATA3 Arbiter

Table 1260: Window0 Control Register (i=0–1)
Offset: Unit0: 0x000A8060, Unit1: 0x000E0060
Offset Formula: 0x00000060+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1260: Window0 Control Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8060, Unit1: 0x000E0060
Offset Formula: 0x00000060+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this window.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. **NOTE:** Must be configured to the SDRAM Controller.
3:2	Reserved	RSVD 0x0	Reserved
1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x0	Window 0 Enable 0 = Disable 1 = Enable

Table 1261: Window0 Base Register (i=0–1)
Offset: Unit0: 0x000A8064, Unit1: 0x000E0064
Offset Formula: 0x00000064+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:4	Base	RW 0x0	Base Address Used with the <Size> field to set the address window size and location. Corresponds to transaction address[47:20].
3:0	Reserved	RSVD 0x0	Reserved

Table 1262: Window0 Size Register (i=0–1)
Offset: Unit0: 0x000A8068, Unit1: 0x000E0068
Offset Formula: 0x00000068+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. The number of 1's specifies the size of the window in 1 MB granularity. **NOTE:** A value of 0x0 specifies 1-MB size.
15:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1263: Window1 Control Register (i=0–1)
Offset: Unit0: 0x000A8070, Unit1: 0x000E0070
Offset Formula: 0x00000070+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	Attr	RW 0x0	Target specific attributes depending on the target interface. See "Window0 Control Register".
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See "Window0 Control Register".
3:2	Reserved	RSVD 0x0	Reserved
1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x0	Window1 Enable See "Window0 Control Register".

Table 1264: Window1 Base Register (i=0–1)
Offset: Unit0: 0x000A8074, Unit1: 0x000E0074
Offset Formula: 0x00000074+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:4	Base	RW 0x0	Base Address See "Window0 Base Register".
3:0	Reserved	RSVD 0x0	Reserved

Table 1265: Window1 Size Register (i=0–1)
Offset: Unit0: 0x000A8078, Unit1: 0x000E0078
Offset Formula: 0x00000078+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size See "Window0 Control Register".
15:0	Reserved	RSVD 0x0	Reserved

Table 1266: Window2 Control Register (i=0–1)
Offset: Unit0: 0x000A8080, Unit1: 0x000E0080
Offset Formula: 0x00000080+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1266: Window2 Control Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8080, Unit1: 0x000E0080
Offset Formula: 0x00000080+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
15:8	Attr	RW 0x0	Target specific attributes depending on the target interface. See "Window0 Control Register".
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See "Window0 Control Register".
3:2	Reserved	RSVD 0x0	Reserved
1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x0	Window2 Enable See "Window0 Control Register".

Table 1267: Window2 Base Register (i=0–1)
Offset: Unit0: 0x000A8084, Unit1: 0x000E0084
Offset Formula: 0x00000084+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:4	Base	RW 0x0	Base Address See "Window0 Base Register".
3:0	Reserved	RSVD 0x0	Reserved

Table 1268: Window2 Size Register (i=0–1)
Offset: Unit0: 0x000A8088, Unit1: 0x000E0088
Offset Formula: 0x00000088+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size See "Window0 Control Register".
15:0	Reserved	RSVD 0x0	Reserved

Table 1269: Window3 Control Register (i=0–1)
Offset: Unit0: 0x000A8090, Unit1: 0x000E0090
Offset Formula: 0x00000090+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	Attr	RW 0x0	Target specific attributes depending on the target interface. See "Window0 Control Register".

Document Classification: Proprietary Information

Table 1269: Window3 Control Register (i=0–1) (Continued)
Offset: Unit0: 0x000A8090, Unit1: 0x000E0090
Offset Formula: 0x00000090+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
7:4	Target	RW 0x0	Specifies the unit ID (target interface) associated with this window. See "Window0 Control Register".
3:2	Reserved	RSVD 0x0	Reserved
1	Reserved	RSVD 0x0	Reserved
0	WinEn	RW 0x0	Window3 Enable See "Window0 Control Register".

Table 1270: Window3 Base Register (i=0–1)
Offset: Unit0: 0x000A8094, Unit1: 0x000E0094
Offset Formula: 0x00000094+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:4	Base	RW 0x0	Base Address See "Window0 Base Register".
3:0	Reserved	RSVD 0x0	Reserved

Table 1271: Window3 Size Register (i=0–1)
Offset: Unit0: 0x000A8098, Unit1: 0x000E0098
Offset Formula: 0x00000098+0xa8000+0x38000*i: where i (0-1) represents Unit

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size See "Window0 Control Register".
15:0	Reserved	RSVD 0x0	Reserved

A.14.1.3 SATA3 Port

Table 1272: Command List Base Address Lower 32-bits Register (i=0–1, n=0–1)
Offset: Unit0 Port0: 0x000A8100, Unit0 Port1: 0x000A8180, Unit1 Port0: 0x000E0100, Unit1 Port1: 0x000E0180
Offset Formula: 0x00000100+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Cmd List Base Addr Low[31:0]	RW 0x0	Command List Base Address Low [31:0]. Lower 32 bits of the Command List Base Address.

Copyright © 2015 Marvell

Table 1273: SATA control register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8100, Unit0 Port1: 0x000A8180, Unit1 Port0: 0x000E0100, Unit1 Port1: 0x000E0180

Offset Formula: 0x00000100+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:24	AHCI_MIN_REV	RW 0x0	AHCI minor revision number
23	SATA_SSU_DIS	RW 0x1	SATA SSU disable
22	SATA_PORT1_EN	RW 0x1	SATA port 1 enable
21	AHCI_AL_PM_EN	RW 0x0	AHCI aggressive link power management enable
20	AHCI_CCC_EN	RW 0x0	AHCI command completion coalescing enable
19:16	SATA_PMACT_DEV_OPT	RW 0x4	SATA port multiplier active device optimization
15	Reserved	RSVD 0x0	Reserved
14	SATA1_PWDN	RW 0x0	SATA 1 power down
13:12	Reserved	RSVD 0x0	Reserved
11	SATA1_PHY_EN	RW 0x0	SATA 1 PHY enable
10:8	SATA1_SPEED_SEL	RW 0x0	SATA 1 speed select 0 = No_Limit: No sata speed limitation, availble 6G/3G/1.5G 1 = Gen1: 1.5G Only 2 = Gen2_or_Gen1: Speed limitation 1.5 or 3G 6 = Gen2: 3G Only 7 = Gen3: 6G Only
7	Reserved	RSVD 0x0	Reserved
6	SATA0_PWDN	RW 0x0	SATA 0 power down
5:4	Reserved	RSVD 0x0	Reserved
3	SATA0_PHY_EN	RW 0x0	SATA 0 PHY enable
2:0	SATA0_SPEED_SEL	RW 0x0	SATA 0 speed select 0 = No_Limit: No SATA speed limitation, 6G/3G/1.5G available 1 = Gen1: 1.5G Only 2 = Gen2_or_Gen1: Speed limitation 1.5 or 3G 6 = Gen2: 3G Only 7 = Gen3: 6G Only

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1274: MBUS request size and interface select register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8104, Unit0 Port1: 0x000A8184, Unit1 Port0: 0x000E0104, Unit1 Port1: 0x000E0184

Offset Formula: 0x00000104+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7	MBUS_REGRET_EN	RW 0x0	Mbus regret enable
6:4	MBUS_RD_SIZE	RW 0x0	Mbus Write Size 0 = Burst_128B
3	MBUS_IN_ORDER	RW 0x0	Mbus in order data transfer
2:0	MBUS_WR_SIZE	RW 0x0	Mbus Read Size 0 = Burst_128B

Table 1275: Command List Base Address Upper 32-bits Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8104, Unit0 Port1: 0x000A8184, Unit1 Port0: 0x000E0104, Unit1 Port1: 0x000E0184

Offset Formula: 0x00000104+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Cmd List Base Addr High	RW 0x0	Command List Base Address High. Upper 32 bits of the Command List Base Address.

Table 1276: SATA memory timing control register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8108, Unit0 Port1: 0x000A8188, Unit1 Port0: 0x000E0108, Unit1 Port1: 0x000E0188

Offset Formula: 0x00000108+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1277: Received FIS Base Address Lower 32-bits Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8108, Unit0 Port1: 0x000A8188, Unit1 Port0: 0x000E0108, Unit1 Port1: 0x000E0188

Offset Formula: 0x00000108+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:8	FIS Base Addr Low	RW 0x000000	FIS Base Address Low. Indicates the 32-bit base physical address for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256-byte aligned as indicated by bits 07:00 being read only. When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.
7:1	Reserved	RO 0x00	Reserved
0	FIS Base Addr Low Dest	RW 0x0	0 = Host: the receive FIS memory is in host side 1 = SRAM: the receive FIS memory is in SRAM

Table 1278: Received FIS Base Address Upper 32-bits Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A810C, Unit0 Port1: 0x000A818C, Unit1 Port0: 0x000E010C, Unit1 Port1: 0x000E018C

Offset Formula: 0x0000010C+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	FIS Base Addr High	RW 0x00000000	FIS Base Address Low[31:0]. Lower 32 bits for the Received FIS Structure.

Table 1279: Port Interrupt Status Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8110, Unit0 Port1: 0x000A8190, Unit1 Port0: 0x000E0110, Unit1 Port1: 0x000E0190

Offset Formula: 0x00000110+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31	CPD Stat	RW1C 0x0	Cold Presence Detect Status. Write a 1 to clear this bit.
30	Task File Err Stat	RW1C 0x0	Task File Error Status. This bit is set whenever the status register is updated by the device and the error bit (bit [0]) is set. Write a 1 to clear this bit.
29	Host Bus Fatal Err Stat	RW1C 0x0	Host Bus Fatal Error Status. Write a 1 to clear this bit.

Copyright © 2015 Marvell

Table 1279: Port Interrupt Status Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8110, Unit0 Port1: 0x000A8190, Unit1 Port0: 0x000E0110, Unit1 Port1: 0x000E0190

Offset Formula: 0x00000110+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
28	Host Bus Data Err Stat	RW1C 0x0	Host Bus Data Error Status. Write a 1 to clear this bit.
27	IF Fatal Err Stat	RW1C 0x0	Interface Fatal Error Status. Write a 1 to clear this bit.
26	IF Non Fatal Stat	RW1C 0x0	Interface Non-Fatal Error Status. Write a 1 to clear this bit.
25	Reserved	RO 0x0	Reserved
24	Overflow Stat	RW1C 0x0	Overflow Status. Write a 1 to clear this bit.
23	IPM Stat	RW1C 0x0	Incorrect Port Multiplier Status. Write a 1 to clear this bit.
22	PHY Ready Ch Stat	RO 0x0	PHY Ready Change Status. When set to 1, indicates the internal PHY_RDY signal-changed state. This reflects the N bit of Serial ATA Error (SError/SCR1) (R130h/R1B0h). Switching in and out of Partial or Slumber mode does not toggle this status.
21:20	Reserved	RO 0x0	Reserved **NOTE:** Do not change the default value.
19	Reserved	RW1C 0x0	Reserved **NOTE:** Do not change the default value.
18	Reserved	RW1C 0x0	Reserved **NOTE:** Do not change the default value.
17	Reserved	RW1C 0x0	Reserved **NOTE:** Do not change the default value.

Copyright © 2015 Marvell

Table 1279: Port Interrupt Status Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8110, Unit0 Port1: 0x000A8190, Unit1 Port0: 0x000E0110, Unit1 Port1: 0x000E0190

Offset Formula: 0x00000110+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
16	Reserved	RW1C 0x0	Reserved **NOTE:** Do not change the default value.
15	Reserved	RW1C 0x0	Reserved **NOTE:** Do not change the default value.
14	Reserved	RW1C 0x0	Reserved **NOTE:** Do not change the default value.
13:8	Reserved	RO 0x0	Reserved **NOTE:** Do not change the default value.
7	Device Mech Pres Stat	RW1C 0x0	Device Mechanical Presence Status. Write a 1 to clear this bit.
6	Port Conn Ch Stat	RO 0x0	Port Connect Change Status. Write a 1 to clear this bit. 0 = Change: Change in current port connect status detected. 1 = No change: No change in current port connect status.
5	Desc Process	RW1C 0x0	Descriptor Processed. Write a 1 to clear this bit.
4	Unkn FIS Int	RO 0x0	Unknown FIS Interrupt. When set to 1, indicates that an unknown FIS was received and copied to the system memory. Write a 1 to clear this bit.
3	Set Device Int	RW1C 0x0	Set Device Bits Interrupt. A Set Device Bits FIS was received with the ‘I’ bit set. This bit can be used to indicate the completion of one or more FPDMA commands. Write a 1 to clear this bit.
2	DMA Set FIS Int	RW1C 0x0	DMA Setup FIS Interrupt. A DMA Setup FIS was received with the ,’I’ bit set. Write a 1 to clear this bit.

Copyright © 2015 Marvell

Table 1279: Port Interrupt Status Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8110, Unit0 Port1: 0x000A8190, Unit1 Port0: 0x000E0110, Unit1 Port1: 0x000E0190

Offset Formula: 0x00000110+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
1	PIO Setup FIS Int	RW1C 0x0	PIO Setup FIS Interrupt. A PIO setup FIS was received with the ,'I' bit set. Write a 1 to clear this bit.
0	D2H Reg FIS Int	RW1C 0x0	Device To Host Register FIS Interrupt. A Device-to-Host Register FIS was received with the 'I' bit set. This bit can be used to indicate the completion of a DMA command or a PIO Data-out command. Write a 1 to clear this bit.

Table 1280: Port Interrupt Enable Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8114, Unit0 Port1: 0x000A8194, Unit1 Port0: 0x000E0114, Unit1 Port1: 0x000E0194

Offset Formula: 0x00000114+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31	CPD En	RW 0x0	Cold Presence Detect Enable. 0 = Disable 1 = Enable
30	Task File Err Int En	RW 0x0	Task File Error Interrupt Enable. 0 = Disable 1 = Enable
29	Host Bus Fatal Err En	RW 0x0	Host Bus Fatal Error Enable. 0 = Disable 1 = Enable
28	Host Bus Data Err En	RW 0x0	Host Bus Data Error Enable. 0 = Disable 1 = Enable
27	IF Fatal Err En	RW 0x0	Interface Fatal Error Enable. 0 = Disable 1 = Enable
26	IF Non Fatal Err En	RW 0x0	Interface Non-Fatal Error Enable. 0 = Disable 1 = Enable
25	Reserved	RO 0x0	25RSVDR0hReserved.
24	Overflow En	RW 0x0	Overflow Enable. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1280: Port Interrupt Enable Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8114, Unit0 Port1: 0x000A8194, Unit1 Port0: 0x000E0114, Unit1 Port1: 0x000E0194

Offset Formula: 0x00000114+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
23	Reserved	RW 0x0	Reserved **NOTE:** Do not change the default value.
22	PHY Ready Ch Int En	RW 0x0	PHY Ready Change Interrupt Enable. Switching in and out of Partial or Slumber mode does not toggle this status.
21:20	Reserved	RO 0x0	Reserved **NOTE:** Do not change the default value.
19	Reserved	RW 0x0	Reserved **NOTE:** Do not change the default value.
18	Reserved	RW 0x0	Reserved **NOTE:** Do not change the default value.
17	Reserved	RW 0x0	Reserved **NOTE:** Do not change the default value.
16	Reserved	RW 0x0	Reserved **NOTE:** Do not change the default value.
15	Reserved	RW 0x0	Reserved **NOTE:** Do not change default value. 0 = Disable 1 = Enable
14	Reserved	RW 0x0	Reserved **NOTE:** Do not change default value. 0 = Disable 1 = Enable
13:8	Reserved	RW 0x0	Reserved **NOTE:** Do not change default value.
7	Device Mech Pres En	RW 0x0	Device Mechanical Presence Enable.
6	Port Conn Ch Stat En	RW 0x0	Port Connect Change Status Enable. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1280: Port Interrupt Enable Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8114, Unit0 Port1: 0x000A8194, Unit1 Port0: 0x000E0114, Unit1 Port1: 0x000E0194

Offset Formula: 0x00000114+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
5	Desc Proc Int En	RW 0x0	Descriptor Processed Interrupt Enable.
4	Unkn FIS Int En	RW 0x0	Unknown FIS Interrupt Enable.
3	Set Device Int En	RW 0x0	Set Device Bits Interrupt Enable.
2	DMA Setup FIS Int En	RW 0x0	DMA Setup FIS Interrupt Enable.
1	PIO Setup FIS Int En	RW 0x0	PIO Setup FIS Interrupt Enable.
0	D2H Reg FIS Int En	RW 0x0	Device-to-Host Register FIS Interrupt Enable.

Table 1281: Port Command and Status Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8118, Unit0 Port1: 0x000A8198, Unit1 Port0: 0x000E0118, Unit1 Port1: 0x000E0198

Offset Formula: 0x00000118+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:28	IF Comm Ctrl	SC 0x0	Interface Communication Control. All other values are reserved. 0 = Idle: No operation / idle 1 = Active state 2 = Partial state 6 = Slumber state
27	ASP	RW 0x0	Aggressive Slumber/Partial State.
26	ALPE	RW 0x0	Aggressive Link Power Management Enable.
25	DLAE	RW 0x0	Drive LED on ATAPI Enable.
24	ATAPI	RW 0x0	Device ATAPI Enable. 0 = Disable: The attached device is not ATAPI. 1 = Enable: The attached device is ATAPI.
23	APSTE	RW 0x0	Automatic Partial to Slumber Transitions Enable. 0 = Disable: The SoC does not perform automatic Partial-to-Slumber state transitions. 1 = Enable: The SoC performs automatic Partial-to-Slumber state transitions.

Copyright © 2015 Marvell

Table 1281: Port Command and Status Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8118, Unit0 Port1: 0x000A8198, Unit1 Port0: 0x000E0118, Unit1 Port1: 0x000E0198

Offset Formula: 0x00000118+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
22	FBSCP	RO 0x1	FIS-Based Switching Capable Port. 0 = No FIS-based switching: The port does not support FIS-based switching. 1 = FIS-based switching: The port supports FIS-based switching.
21	ESP	RO 0x0	External SATA Port Available. 0 = No external SATA port: External SATA port is not available. 1 = External SATA port: External SATA port is available.
20	CPD Detect Supp	RO 0x0	Cold Presence Detection Support. 0 = Detection not supported: The platform does not support cold presence detection. 1 = Detection supported: The platform supports cold presence detection.
19	MPSP	RO 0x0	Mechanical Presence Switch Attached to Port 0 = No switch support: The platform does not support a mechanical presence switch. 1 = Switch support: The platform supports a mechanical presence switch.
18	HPCP	RO 0x0	Hot Plug Capable Port. 0 = Not externally accessible: The port's signal and power connectors are not externally accessible. 1 = Externally accessible: The port's signal and power connectors are externally accessible.
17	PM Attached	RW 0x0	Port Multiplier Attached. This bit is read/write for SoCs that support a port multiplier. Software is responsible for detecting whether a port multiplier is present. The hardware does not auto-detect the presence of a port multiplier.
16	CPS	RO 0x0	Cold Presence State. 0 = Device not detected: Cold presence does not detect a device on this port. 1 = Device detected: Cold presence detects a device on this port.
15	CR	RO 0x0	Command List Running. When this bit is set, the command list engine for the port is running, regardless of the DMA or PIO protocol.
14	FR	RO 0x0	FIS Receive Running. When this bit is set, the FIS receive DMA is running.
13	MPSS	RO 0x0	Mechanical Presence Switch State.
12:8	Current Cmd Slot	RW 0x0	Current Command Slot. The command slot value of the command that is currently running on the SoC.
7:5	Reserved	RW 0x0	Reserved. **NOTE:** Do not change default value.
4	FIS Rx En	RW 0x0	FIS Receive Enable. When set, the controller posts received FIS data into the FIS receive area pointed to by the FIS Base Address.
3	CLO	RW 0x0	Command List Override.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1281: Port Command and Status Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8118, Unit0 Port1: 0x000A8198, Unit1 Port0: 0x000E0118, Unit1 Port1: 0x000E0198

Offset Formula: 0x00000118+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
2	POD Support	RW 0x0	Power On Device Support.
1	Reserved	RSVD 0x0	Reserved
0	Cmd Start	RW 0x0	Command Start. When set, the SoC processes the command list. When cleared, the SoC stops processing the command list. Whenever this bit is changed from a 0 to a 1, the SoC starts processing the command list at entry 0. Whenever this bit is changed from a 1 to a 0, the Command Issue register is cleared by the SoC upon the SoC putting the controller into an Idle state.

Table 1282: Reserved 1 Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A811C, Unit0 Port1: 0x000A819C, Unit1 Port0: 0x000E011C, Unit1 Port1: 0x000E019C

Offset Formula: 0x0000011C+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RW 0x0	Reserved **NOTE:** Do not change default value.

Table 1283: Port Task File Data Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8120, Unit0 Port1: 0x000A81A0, Unit1 Port0: 0x000E0120, Unit1 Port1: 0x000E01A0

Offset Formula: 0x00000120+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RO 0x0	Reserved
15:8	Err Reg	RO 0x0	Error. Contains the latest copy of the task file error register.
7:0	Status Reg	RO 0x7f	Status. Contains the latest copy of the task file status register.

Copyright © 2015 Marvell

Table 1284: Port Signature Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8124, Unit0 Port1: 0x000A81A4, Unit1 Port0: 0x000E0124, Unit1 Port1: 0x000E01A4

Offset Formula: 0x00000124+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
This is a 32-bit register which contains the initial signature of an attached device when the first D2H Register FIS is received from that device. It is updated once after a reset sequence.			
31:24	LBA High	RO 0xFF	LBA High.
23:16	LBA Mid	RO 0xff	LBA Medium.
15:8	LBA Low	RO 0xff	LBA Low.
7:0	Sector Cnt	RO 0xff	Sector Count.

Table 1285: SATA Status (SStatus/SCR0) Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8128, Unit0 Port1: 0x000A81A8, Unit1 Port0: 0x000E0128, Unit1 Port1: 0x000E01A8

Offset Formula: 0x00000128+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
NOTE: See the Serial-ATA specification for a detailed description.			
31:12	Reserved	RSVD 0x0	Reserved
11:8	IPM	RO 0x0	These bits set the current interface power management state. All other values are reserved. 0 = No communication: Device not present, or communication not established. 1 = Active: Interface in active state. 2 = Partial: Interface in PARTIAL power management state. 6 = Slumber: Interface in SLUMBER power management state.
7:4	SPD	RO 0x0	These bits set whether the negotiated interface communication speed is established. All other values are reserved. 0 = No negotiation: No negotiated speed. The device is not present or communication is not established. 1 = Gen1 negotiation: Generation 1 communication rate negotiated. 2 = Gen2 negotiation: Generation 2 communication rate negotiated. 3 = Gen3 negotiation: Generation 3 communication rate negotiated.

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1285: SATA Status (SStatus/SCR0) Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8128, Unit0 Port1: 0x000A81A8, Unit1 Port0: 0x000E0128, Unit1 Port1: 0x000E01A8

Offset Formula: 0x00000128+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
3:0	DET	RO 0x0	These bits set the interface device detection and PHY state. All other values are reserved. 0 = No communication: No device detected, and PHY communication is not established. 1 = Device with no comm: Device presence detected, but PHY communication is not established. 3 = Device with comm: Device presence detected, and PHY communication is established. 4 = PHY offline: PHY in offline mode as a result of the interface being disabled or running in a loopback mode.

Table 1286: SATA Control (SControl/SCR2) Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A812C, Unit0 Port1: 0x000A81AC, Unit1 Port0: 0x000E012C, Unit1 Port1: 0x000E01AC

Offset Formula: 0x0000012C+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
This register is also known as SCR2, or SControl.			
31:20	Reserved	RO 0x0	Reserved
19:16	Reserved	RO 0x0	Reserved
15:12	Reserved	RO 0x0	Reserved
11:8	IPM SCR2	RW 0x0	Interface Power Management (IPM). field represents the enabled interface power management states that can be invoked via the Serial-ATA interface power management capabilities 0 = No restrictions: No interface power management state restrictions. 1 = Partial: Transition to the Partial power management state disabled. 2 = Slumber: Transition to the Slumber power management state disabled. 3 = Partial/Slumber: Transition to both the Partial and Slumber power management states disabled.
7:4	Speed SCR2	RW 0x0	Interface Speed Allowed (SPD). SPD field represents the highest allowed communication speed the interface is able to negotiate. Value of this register also depended by sata phy control register (access by vendor unique register) bits 3 , 23 for channel 0 and 11,22, 23 for channel 1 0 = No rerstriction: No speed negotiation restrictions. 1 = Generation 1: Limit speed negotiation to a rate not greater than Generation 1 (SATA 1.5 Gbps) communication rate. 2 = Generation 2: Limit speed negotiation to a rate not greater than Generation 2 (SATA 3.0 Gbps) communication rate. 3 = Generation 3: Limit speed negotiation to a rate not greater than Generation 3 (SATA 6.0 Gbps) communication rate.

Copyright © 2015 Marvell

Table 1286: SATA Control (SControl/SCR2) Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A812C, Unit0 Port1: 0x000A81AC, Unit1 Port0: 0x000E012C, Unit1 Port1: 0x000E01AC

Offset Formula: 0x0000012C+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
3:0	Detect SCR2	RW 0x4	Device Detection Initialization (DET). DET field controls the host adapter device detection and interface initialization. Value of this register also depended by sata phy control register (access by vendor unique register) bits 3 , 23 for channel 0 , bits 11,22,23 for channel 1 and command register 0x118 for channel 0 bit 1 ,0x198 bit 1 for channel 1 0 = No request: No device detection or initialization action requested. 1 = Interface comm: Perform interface communication initialization sequence to establish communication. 2 = Disable PHY: Disable the Serial-ATA interface and put the PHY in offline mode.

Table 1287: SATA Error (SError/SCR1) Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8130, Unit0 Port1: 0x000A81B0, Unit1 Port0: 0x000E0130, Unit1 Port1: 0x000E01B0

Offset Formula: 0x00000130+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
This register is also known as SCR1, or SError.			
31:27	Reserved	RO 0x0	Reserved **NOTE:** Do not change the default value.
26	Dev Presence Exch	RW1C 0x0	Device Presence Change Detected. When set to 1 this bit indicates that device presence has changed since the last time this bit was cleared. The means by which the implementation determines that the device presence has changed is vendor specific. This bit may be set anytime a PHY reset initialization sequence occurs as determined by reception of the COMINIT signal whether in response to: - A new device being inserted OR - A COMRESET having been issued OR - Power-up
25	Unknown FIS Type	RW1C 0x0	Unknown FIS Type Error Detected.
24	TPort Trans Err	RW1C 0x0	Transport Transition Error Detected. When set to 1, this bit indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared.
23	Link Seq Err	RW1C 0x0	Link Sequence Error Detected. When set to 1, this bit indicates that one or more Link state machine error conditions was encountered since the last time this bit was cleared. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition.

Copyright © 2015 Marvell

Table 1287: SATA Error (SError/SCR1) Register (i=0–1, n=0–1) (Continued)

Offset: Unit0 Port0: 0x000A8130, Unit0 Port1: 0x000A81B0, Unit1 Port0: 0x000E0130, Unit1 Port1: 0x000E01B0

Offset Formula: 0x00000130+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
22	Handshake Err	RW1C 0x0	Handshake Error Detected. When set to one, this bits indicates that one or more R_ERR handshake responses was received in response to frame transmission. Such errors may be the result of: - A CRC error detected by the recipient - A disparity or 10-bit to 8-bit decoding error - Other error conditions, leading to a negative handshake on a transmitted frame
21	CRC Err	RW1C 0x0	CRC Error Detected. When set to 1, this bit indicates that one or more CRC errors occurred with the Link Layer since the bit was last cleared.
20	Disp Err	RW1C 0x0	Disparity Error Detected. When set to 1, this bit indicates that incorrect disparity was detected one or more times since the last time the bit was cleared.
19	10B_8B Dec Err	RW1C 0x0	10B to 8B Decode Error Detected. When set to 1, this bit indicates that one or more 10-bit to 8-bit decoding errors occurred since the bit was last cleared.
18	Com Wake Err	RW1C 0x0	COMWAKE Error Detected. When set to 1, this bit indicates that a Comm Wake signal was detected by the PHY since the last time this bit was cleared.
17	Reserved	RO 0x0	Reserved
16	PHY Ready Chnage	RW1C 0x0	PHY Ready Change Detected. When set to 1, this bit indicates that the PhyRdy changed state since the last time this bit was cleared.
15:2	Reserved	RO 0x0	Reserved
1	Recovered Communication Error	RW 0x0	Communication between the device and host was temporarily lost but was re-established. This can arise from: - A device temporarily being removed. - From a temporary loss of PHY synchronization. - From other causes and may be derived from the PhyNRdy signal between the PHY and Link layers. No action is required by the host software since the operation ultimately succeeded. However, the host software may elect to track such recovered errors to gauge overall communications integrity and potentially step-down the negotiated communication speed.
0	Reserved	RO 0x0	Reserved

Table 1288: SATA Active (SActive/SCR3) Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8134, Unit0 Port1: 0x000A81B4, Unit1 Port0: 0x000E0134, Unit1 Port1: 0x000E01B4

Offset Formula: 0x00000134+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Device Status SCR3	RW 0x0	Device Status. System software sets this field for SATA queuing operations prior to setting the Command issue register bit in the same command slot entry. This field is cleared using Set Device Bits FIS. This field is also cleared when the Start bit of the Command register is cleared from a 1 to a 0 by the software. This field is not cleared by a COMRESET or a software reset.

Table 1289: Command Issue Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8138, Unit0 Port1: 0x000A81B8, Unit1 Port0: 0x000E0138, Unit1 Port1: 0x000E01B8

Offset Formula: 0x00000138+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Cmd Issue	RW 0x0	Commands Issued. This field is set by software to indicate to the SoC that a command was built into the system memory for a command slot, and might be sent to the device. When the SoC receives a FIS that clears BSY, DRQ, and ERR bits for the command, the corresponding bit in this register for that command slot is also cleared. This field is also cleared when the Start bit of the Command register is cleared from a 1 to a 0 by the firmware.

Table 1290: Port SATA Notification (SNotification/SCR4) Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A813C, Unit0 Port1: 0x000A81BC, Unit1 Port0: 0x000E013C, Unit1 Port1: 0x000E01BC

Offset Formula: 0x0000013C+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RO 0x0	Reserved
15:0	PM Notify	RW1C 0x0	Port Multiplier Notification.

Copyright © 2015 Marvell

Table 1291: Port FIS-Based Switching Control Register (i=0–1, n=0–1)

Offset: Unit0 Port0: 0x000A8140, Unit0 Port1: 0x000A81C0, Unit1 Port0: 0x000E0140, Unit1 Port1: 0x000E01C0

Offset Formula: 0x00000140+0x80*n + 0xa8000+0x38000*i : where i (0-1) represents Unit, where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:20	Reserved	RO 0x0	Reserved.
19:16	PM Device W Err	RO 0x0	Port Multiplier Device with Error. The value of this field indicates the port multiplier port that had a fatal error.
15:12	Act Device Opt	RO 0x4	Active Device Optimization. This field indicates the number of optimized FIS-based switching devices.
11:8	PM Device To Issue	RW 0x0	Port Multiplier Device to Issue. The software programs the value of the Port Multiplier port for the next issued command.
7:3	Reserved	RO 0x0	Reserved
2	Single Device Err	RO 0x0	Single Device Error. When this bit is 1, a fatal error has occurred.
1	Device Err Clr	RW1C 0x0	Device Error Clear. After error recovery, this bit is cleared to 0h.
0	PM En	RW 0x0	Port Multiplier Enable. 0 = Disable: FIS-based switching is not used. 1 = Enable: A Port Multiplier is in use and FIS-based switching is used for communication.

A.15 Audio (I2S / S/PDIF) Registers

The following table provides a summarized list of all registers that belong to the Audio (I2S / S/PDIF), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1292: Summary Map Table for the Audio (I2S / S/PDIF) Registers

Register Name	Offset	Table and Page
Audio Address Window Control		
Recording Window Base Address Register	0x000E8A00	Table 1293, p. 1055
Recording Window Control Register	0x000E8A04	Table 1294, p. 1055
Playback Window Base Address Register	0x000E8A08	Table 1295, p. 1056
Playback Window Control Register	0x000E8A0C	Table 1296, p. 1056
Audio Clocking Control		
Sample Counters Control Register	0x000E9220	Table 1297, p. 1057
Playback Sample Counter Register	0x000E9224	Table 1298, p. 1057
Recording Sample Counter Register	0x000E9228	Table 1299, p. 1058
Clocks Control Register	0x000E9230	Table 1300, p. 1058
Reserved Register	0x000E9238	Table 1301, p. 1058
Audio Interrupt and Error		
Audio Error Cause Register	0x000E9300	Table 1302, p. 1059
Audio Error Mask Register	0x000E9304	Table 1303, p. 1059
Audio Interrupt Cause Register	0x000E9308	Table 1304, p. 1059
Audio Interrupt Mask Register	0x000E930C	Table 1305, p. 1061
Recorded Byte Count for Interrupt Register	0x000E9310	Table 1306, p. 1061
Playback Byte Count for Interrupt Register	0x000E9314	Table 1307, p. 1061
Playback Byte Counter for Interrupt Register	0x000E9318	Table 1308, p. 1061
Recorded Byte Counter for Interrupt Register	0x000E931C	Table 1309, p. 1062
Audio Playback		
Playback Control Register	0x000E9100	Table 1310, p. 1062
Playback Start Address Register	0x000E9104	Table 1311, p. 1065
Playback Buffer Size Register	0x000E9108	Table 1312, p. 1065
Playback Buffer Byte Counter Register	0x000E910C	Table 1313, p. 1065
SPDIF Playback Control Register	0x000EA204	Table 1314, p. 1066
SPDIF Playback Channel Status Left n Register (n=0–5)	status0: 0x000EA280, status1: 0x000EA284, status2: 0x000EA288, status3: 0x000EA28C, status4: 0x000EA290, status5: 0x000EA294	Table 1315, p. 1067
SPDIF Playback Channel Status Right n Register (n=0–5)	status0: 0x000EA2A0, status1: 0x000EA2A4, status2: 0x000EA2A8, status3: 0x000EA2AC, status4: 0x000EA2B0, status5: 0x000EA2B4	Table 1316, p. 1067

Copyright © 2015 Marvell

Table 1292: Summary Map Table for the Audio (I2S / S/PDIF) Registers (Continued)

Register Name	Offset	Table and Page
SPDIF Playback User Bits Left n Register (n=0–5)	User0: 0x000EA2C0, User1: 0x000EA2C4, User2: 0x000EA2C8, User3: 0x000EA2CC, User4: 0x000EA2D0, User5: 0x000EA2D4	Table 1317, p. 1067
SPDIF Playback User Bits Right n Register (n=0–5)	User0: 0x000EA2E0, User1: 0x000EA2E4, User2: 0x000EA2E8, User3: 0x000EA2EC, User4: 0x000EA2F0, User5: 0x000EA2F4	Table 1318, p. 1067
I2S Playback Control Register	0x000EA508	Table 1319, p. 1068
Audio Recording		
Recording Control Register	0x000E9000	Table 1320, p. 1068
Recording Start Address Register	0x000E9004	Table 1321, p. 1070
Recording Buffer Size Register	0x000E9008	Table 1322, p. 1070
Recording Buffer Byte Counter Register	0x000E900C	Table 1323, p. 1071
I2S Recording Control Register	0x000EA408	Table 1324, p. 1071

A.15.1 Audio Address Window Control

Table 1293: Recording Window Base Address Register
Offset: 0x000E8A00

Bit	Field	Type/InitVal	Description
31:16	Recording Base	RW 0x0	Window Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 1294: Recording Window Control Register
Offset: 0x000E8A04

Bit	Field	Type/InitVal	Description
31:16	Recording Win Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by a sequence of 0s. The number of 1s specifies the size of the window in 64 KByte granularity (e.g., a value of 0x00FF specifies 16 MByte). **NOTE:** The Window Size must be an integer multiple of the <Recording DMA Burst Size> in the Recording Control Register.
15:8	Recording Attr	RW 0xE	Attribute

Copyright © 2015 Marvell

Table 1294: Recording Window Control Register (Continued)
Offset: 0x000E8A04

Bit	Field	Type/InitVal	Description
7:4	Recording TargetID	RW 0x0	Target Unit ID
3:1	RSVD	RSVD 0x0	Reserved
0	RSVD	RW 0x1	Always write 1'b1 to this bit.

Table 1295: Playback Window Base Address Register
Offset: 0x000E8A08

Bit	Field	Type/InitVal	Description
31:16	Playback Base	RW 0x1000	Window Base Address
15:0	Reserved	RSVD 0x0	Reserved

Table 1296: Playback Window Control Register
Offset: 0x000E8A0C

Bit	Field	Type/InitVal	Description
31:16	Playback Size	RW 0x0FFF	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by a sequence of 0s. The number of 1s specifies the size of the window in 64 KByte granularity (e.g., a value of 0x00FF specifies 16 MByte). **NOTE:** The Window Size must be an integer multiple of the <Playback DMA Burst Size> in the Playback Control Register.
15:8	Playback Attr	RW 0xD	Attribute
7:4	Playback TargetID	RW 0x0	Target Unit ID
3:1	RSVD	RSVD 0x0	Reserved
0	RSVD	RW 0x1	Always write 1'b1 to this bit.

Copyright © 2015 Marvell

A.15.2 Audio Clocking Control

Table 1297: Sample Counters Control Register
Offset: 0x000E9220

Bit	Field	Type/InitVal	Description
31:10	Reserved	RSVD 0x0	Reserved
9	Clear Playback Counter	WO 0x0	When set together with bit [8], both counters will be cleared on the same cycle.
8	Clear Recording Counter	WO 0x0	When set together with bit [9], both counters will be cleared on the same cycle.
7:2	Reserved	RSVD 0x0	Reserved
1	Activate Playback Counter	RW 0x0	When set together with bit [0], counters will start counting on the same cycle. 0 = Stop Counter 1 = Activate Counter
0	Activate Recording Counter	RW 0x0	When set together with bit [1], counters will start counting on the same cycle. 0 = Stop Counter 1 = Activate Counter

Table 1298: Playback Sample Counter Register
Offset: 0x000E9224

Bit	Field	Type/InitVal	Description
31:0	Playback Sample Counter	RO 0x0	Incremented by 1 after each left channel and right channel pair has been transmitted. Notes: 1.This counter is a wrap around counter. 2. If SPDIF playback is active, then this counter counts SPDIF samples; otherwise, I2S outgoing samples are counted. In the case of simultaneous I2S and SPDIF playback, samples are sent to both interfaces at the same frequency.

Copyright © 2015 Marvell

Table 1299: Recording Sample Counter Register
Offset: 0x000E9228

Bit	Field	Type/InitVal	Description
31:0	Recorded Sample Counter	RO 0x0	Incremented by 1 for each incoming left channel and right channel pair. In Mono mode, only one of the channels is recorded so the counter reflects the number of recorded samples. In Stereo mode, the counter reflects the number of stereo samples that were received. **NOTE:** This counter is a wrap around counter.

Table 1300: Clocks Control Register
Offset: 0x000E9230

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	RSVD	RW 0x0	Always write 0 to this bit.
14:2	Reserved	RSVD 0x0	Reserved
1:0	MCLK source	RW 0x0	MCLK = 256 Fs, where Fs = 44.1 kHz, 48 kHz, 96 kHz. **NOTE:** It is possible to move between DCO mclk and SPCR mclk "on the fly" with no hardware glitches. Selection of External mclk or a move from the external mclk to another mclk source should be done when playback is not active. 0 = DCO: MCLK is from DCO 2 = SPCR: MCLK is from SPCR 3 = External: MCLK is from external source

Table 1301: Reserved Register
Offset: 0x000E9238

Bit	Field	Type/InitVal	Description
31:1	Reserved	RW 0x7FFF8000	Reserved
0	RSVD	RW 0x1	Always write 1 to this bit.

Copyright © 2015 Marvell

A.15.3 Audio Interrupt and Error

Table 1302: Audio Error Cause Register
Offset: 0x000E9300

Bit	Field	Type/InitVal	Description
All bits are RW1C only. A cause bit sets upon an event occurrence. A write of 1 clears the bit. A write of 0 has no effect.			
31:7	Reserved	RSVD 0x0	Reserved
6	Playback SPDIF Underrun Err	RW1C 0x0	Set when an underrun error occurs during SPDIF playback. The DMA Engine fails to supply playback data on time.
5	Playback I2S Underrun Err	RW1C 0x0	Set when an underrun error occurs during I2S playback. The DMA Engine fails to supply playback data on time.
4	Recording Overrun Err	RW1C 0x0	Set when an overrun error occurs during recording. The DMA Engine fails to transfer recording data to memory before new audio data arrives.
3	Playback Data Err	RW1C 0x0	Set when the playback data has a parity error indication.
2	Playback Addr Miss	RW1C 0x0	Set when the playback buffer address misses the BAR.
1	Reserved	RSVD 0x0	Reserved
0	Recording Addr Miss	RW1C 0x0	Set when the recording buffer address misses the BAR.

Table 1303: Audio Error Mask Register
Offset: 0x000E9304

Bit	Field	Type/InitVal	Description
31:0	AudioErrorMask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the Audio Error interrupt.

Table 1304: Audio Interrupt Cause Register
Offset: 0x000E9308

Bit	Field	Type/InitVal	Description
Most of the bits are RW1C only. A cause bit sets upon an event occurrence. A write of 1 clears the bit. A write of 0 has no effect.			
31:22	Reserved	RSVD 0x0	Reserved
21	spdif_rc_locked	RW1C 0x0	When asserted indicates a change of <spdif_rc_lock_status> from unlocked to locked. Current lock status is indicated at <spdif_rc_lock_status>

Copyright © 2015 Marvell

Table 1304: Audio Interrupt Cause Register (Continued)
Offset: 0x000E9308

Bit	Field	Type/InitVal	Description
20:15	Reserved	RSVD 0x0	Reserved
14	Played num of Bytes Int	RW1C 0x0	This interrupt is set whenever the value in the <Playback Byte Counter for Interrupt> field of the Playback Byte Counter for Interrupt Register reaches the value that was configured in the <Playback Byte Count for Interrupt> field of the Playback Byte Count for Interrupt Register.
13	Recorded num of Bytes Int	RW1C 0x0	This interrupt is set whenever the value in the <Recorded Byte Counter for Interrupt> field of the Recorded Byte Counter for Interrupt Register reaches the value that was configured in the <Recorded Byte Count for Interrupt> field of the Recorded Byte Count for Interrupt Register.
12	Reserved	RSVD 0x0	Reserved
11	spdif_pb_block_start	RW1C 0x0	Indicates start of SPDIF playback block. **NOTE:** When status/user bits are read from the register file, they may be changed before the first operation and after each start of block interrupt. If status/user bits need to be changed in the next block, then the user has the time it takes to play one block, to update the new status/user registers.
10:8	Reserved	RSVD 0x0	Reserved
7	DMA Playback Buf End	RW1C 0x0	Set by the hardware when the playback DMA engine reaches the end of the playback buffer. Firmware should write new playback data into this quarter/half of the cyclic buffer.
6	DMA Playback 3rd Qtr Buf	RW1C 0x0	Set by the hardware when the playback DMA engine reaches the end of the third quarter in the playback buffer. Firmware should write new playback data into this quarter of the cyclic buffer.
5	DMA Playback Half Buf	RW1C 0x0	Set by the hardware when the playback DMA engine reaches half of the playback buffer. Firmware should write new playback data into this quarter/half of the cyclic buffer.
4	DMA Playback 1st Qtr Buf	RW1C 0x0	Set by the hardware when the playback DMA engine reaches the first quarter of the playback buffer. Firmware should write new playback data into this first quarter of the cyclic buffer.
3	DMA Recording End Buf	RW1C 0x0	Set by the hardware when the recording DMA engine reaches the end of the recording buffer. Firmware should move this newly recorded data from this end of the cyclic buffer to another memory location to avoid loosing the data when the Recording DMA re-writes this quarter/half of the cyclic buffer.
2	DMA Recording 3rd Qtr Buf	RW1C 0x0	Set by the hardware when the recording DMA engine reaches the end of the third quarter in the recording buffer. Firmware should move this newly recorded data from this quarter of the cyclic buffer to another memory location to avoid loosing the data when the Recording DMA re-writes this quarter of the cyclic buffer.
1	DMA Recording Half Buf	RW1C 0x0	Set by the hardware when the recording DMA engine reaches half of the recording buffer. Firmware should move this newly recorded data from this quarter/half of the cyclic buffer to another memory location to avoid loosing the data when the Recording DMA re-writes this quarter/half of the cyclic buffer.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 1304: Audio Interrupt Cause Register (Continued)
Offset: 0x000E9308

Bit	Field	Type/InitVal	Description
0	DMA Recording 1st Qtr Buf	RW1C 0x0	Set by the hardware when the recording DMA engine reaches the end of the first quarter in the recording buffer. Firmware should move this newly recorded data from this quarter of the cyclic buffer to avoid loosing the data when the Recording DMA re-writes this quarter of the cyclic buffer.

Table 1305: Audio Interrupt Mask Register
Offset: 0x000E930C

Bit	Field	Type/InitVal	Description
31:0	AudioInterruptMask	RW 0x0	Mask bit per cause bit. If a bit is set to 1, the corresponding event is enabled. Mask does not affect setting of the Interrupt Cause register bits; it only affects the assertion of the Audio interrupt. Always write 0 to bits [15], [20], [24], and [25].

Table 1306: Recorded Byte Count for Interrupt Register
Offset: 0x000E9310

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:0	Recorded Byte Count for Interrupt	RW 0x800000	An interrupt is asserted whenever the configured number of bytes has been recorded. **NOTE:** A value of 0 is reserved.

Table 1307: Playback Byte Count for Interrupt Register
Offset: 0x000E9314

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:0	Playback Byte Count for int	RW 0x800000	An interrupt is asserted whenever the configured number of bytes has been played. **NOTE:** A value of 0 is reserved.

Table 1308: Playback Byte Counter for Interrupt Register
Offset: 0x000E9318

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1308: Playback Byte Counter for Interrupt Register (Continued)
Offset: 0x000E9318

Bit	Field	Type/InitVal	Description
23:0	Playback Byte Counter for Interrupt	RO 0x0	Number of bytes that were transferred by the playback DMA from playback enable or since the last Played-num-of-Bytes-Int interrupt assertion. This counter is reset when the counter reaches the value set in the Playback Byte Count for Interrupt register or when playback is disabled.

Table 1309: Recorded Byte Counter for Interrupt Register
Offset: 0x000E931C

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:0	Recorded Byte Counter for Interrupt	RO 0x0	Number of bytes that were transferred by the recording-DMA from recording-enable or since the last Recorded-num-of-Bytes-Int interrupt assertion. **NOTE:** This counter is reset when its value reaches the value in the <Recorded Byte Count for Interrupt> field of the Recorded Byte Count for Interrupt Register, or when recording is re-enabled.

A.15.4 Audio Playback

Table 1310: Playback Control Register
Offset: 0x000E9100

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	Playback Busy	RO 0x0	When asserted: playback is on going. **NOTE:** After either I2S or SPDIF playback is disabled, this bit must be read twice as "idle" before any "Audio Playback" register can be changed. 0 = DMA is idle 1 = DMA is active
15:13	Reserved	RSVD 0x0	Reserved
12:11	Playback DMA Burst Size	RW 0x2	Specifies the Burst Size of the DMA. 1 = 32 Bytes 2 = 128 Bytes 3 = Reserved
10	Loop Back	RW 0x0	When Loopback is enabled, playback data is looped back to be recorded. Loopback mode is used for test. 0 = Normal: Normal operation 1 = Loopback: Loopback test mode is enabled.

Copyright © 2015 Marvell

Table 1310: Playback Control Register (Continued)
Offset: 0x000E9100

Bit	Field	Type/InitVal	Description
9	Playback Pause	RW 0x0	When this bit is set, the playback DMA stops. Pause differs from mute, in which the DMA playback operation is on going. **NOTE:** <Playback Pause> must not be disabled when <SPDIF Playback Enable> or <I2S Playback Enable> are deasserted. After playback is disabled, <Playback Pause> must remain active until <Playback Busy> is read twice as IDLE. 0 = Pause disabled 1 = Pause enabled
8	SPDIF Playback Mute	RW 0x0	When this bit is set, the DMA playback continues, but playback data is not transmitted. Mute is different from Pause since DMA playback stops while pausing. In normal operation, this bit must be deasserted. **NOTE:** 1.If the validity bit is taken from the <SPDIF PB Reg Validity> bit of the SPDIF Playback Control Register, then before mute begins and until it ends, it is recommended to assert the <SPDIF PB Reg Validity>. 2.<SPDIF Playback Mute> must not be disabled when <SPDIF Playback Enable> is deasserted. After Playback is disabled, <SPDIF Playback Mute> must remain active until <Playback Busy> is read twice as IDLE. 0 = Mute disabled 1 = Mute enabled
7	I2S Playback Mute	RW 0x0	When this bit is set, the DMA playback continues, but playback data is not transmitted. Mute is different from Pause: during Pause the DMA playback stops. In normal operation, this bit must be de-asserted. **NOTE:** <I2S Playback Mute> must not be disabled when <I2S Playback Enable> is deasserted. After playback is disabled, <I2S Playback Mute> must remain active until <Playback Busy> is read twice as IDLE. 0 = Mute disabled 1 = Mute enabled
6:5	Playback Mono	RW 0x0	When the Mono mode is used, the ADU Engine treats playback data as mono data. **NOTE:** 1. If only one of the channels is used in Mono (data is not duplicated to both channels) mode, then the unused channel data is zero. 2. If recorded data does not contain block start information. For example, since only the right channel was recorded, then user bits cannot be taken from memory if they are required to align to the start of block. 3. If the SPDIF start of block information should be taken from memory (since user bits are taken from memory and should align to the start of block) then values 2 and 3 cannot be used, i.e., only Stereo mode and Left Channel Mono mode are permissible. 4. If stereo data is in the playback buffer and the playback is in Mono mode, the playback engine treats the data as if it belonged to a single channel. Therefore, the audio is played at half the speed, and lasts twice as long. 0 = Stereo Mode: Mono mode is off. 1 = Left Channel Mono: Left channel is used in Mono playback mode. 2 = Right Channel Mono: Right channel is used in Mono playback mode. 3 = Both Channels Mono: Same playback data is duplicated to both channels.

Copyright © 2015 Marvell

Table 1310: Playback Control Register (Continued)
Offset: 0x000E9100

Bit	Field	Type/InitVal	Description
4	SPDIF Playback Enable	RW 0x0	Used to enable/disable SPDIF playback and the playback DMA. **NOTE:** It is possible to simultaneously playback the same data through I2S and SPDIF. In this case, both interfaces should be enabled and disabled concurrently. Use <SPDIF Playback Mute> or <I2S Playback Mute> if only one interface should remain active, after both interfaces are simultaneously enabled. 0 = SPDIF playback disabled 1 = SPDIF playback enabled
3	I2S Playback Enable	RW 0x0	Used to enable/disable playback DMA, as well as I2S data, I2S bit clock, and I2S left/right clock. **NOTE:** It is possible to simultaneously playback the same data through I2S and SPDIF. In this case, both interfaces should be enabled and disabled concurrently. Use <SPDIF Playback Mute> or <I2S Playback Mute> if only one interface should remain active, after both interfaces are simultaneously enabled. 0 = I2S playback disabled 1 = I2S playback enabled
2:0	Playback Sample Size	RW 0x1	This field determines the memory structure during playback. If playback data was recorded using the Audio unit then the value of this field should be the same as the value of the <Recording Sample Size> field in the Recording Control Register, that was used in the recording. **NOTE:** 1. The SPDIF playback only supports up to 24 bits. 2. When I2S is played, always configure in addition, the <I2S PB Sample Size> in the I2S Playback Control Register to the sample size of the real audio stream. 0 = 32 bits: I2S mode only. Reserved for SPDIF mode. Each 32-bit word in memory includes 32 bits of audio data. 1 = 24 bits: Each 32-bit word in memory includes 24 bits of audio data at bits [23:0]. If this mode is used for a sampling size of less than 24 bits, then bit [23] should still hold the MS bit of the data. Redundant LS bits should be 0. 2 = 20 bits: Each 32-bit word in memory includes 20 bits of audio data at bits [19:0]. 3 = 16 bits compact: Each 32-bit word in memory consists of two halfwords of 16 bits of audio data. NOTE: Using this mode for SPDIF means that user and status bits are stored in the internal buffers and cannot be stored with the data. 7 = 16 bits: Each 32-bit word in memory includes 16 bits of audio data at bits [15:0].

Copyright © 2015 Marvell

Table 1311: Playback Start Address Register
Offset: 0x000E9104

Bit	Field	Type/InitVal	Description
31:0	Playback Buffer Start Address	RW 0x0	Contains the pointer to the starting location of the playback buffer in memory. Data in that location will be the first data to be played. **NOTE:** 1. The Base Address must be 8-byte aligned. 2. For less DMA memory bandwidth consumption, it is highly recommended that the address be aligned to the <Playback DMA Burst Size> as configured in the Playback Control Register.

Table 1312: Playback Buffer Size Register
Offset: 0x000E9108

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved
21:0	Playback Buffer Size	RW 0x0	Contains the size of the playback buffer in Word units (32 bit) minus one. If set to 0, playback buffer is 1 Word (4 bytes total). If set to 1, playback buffer is 2 Words (8 bytes total). . . . If set to 0x3FFFFF (max allowed value), playback buffer size is 4M words or 16 MBytes. **NOTE:** Playback Buffer Size must be an integer multiple of <Playback DMA Burst Size>, as determined in the Playback Control register. The buffer size should be greater than the burst size.

Table 1313: Playback Buffer Byte Counter Register
Offset: 0x000E910C

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:0	Playback buf Byte Counter	RO 0x0	Number of bytes that were transferred by the playback DMA. Modulates the playback buffer size. **NOTE:** This counter is reset when its value reaches the value of the <Playback Buffer Size> field in the Playback Buffer Size Register or when playback is disabled.

Copyright © 2015 Marvell

Table 1314: SPDIF Playback Control Register
Offset: 0x000EA204

Bit	Field	Type/InitVal	Description
31:20	RSVD	RW 0x0	Always write 0 to this field.
19:18	RSVD	RW 0x0	Always write 0 to this field.
17	SPDIF PB Non PCM	RW 0x0	Defines the audio stream type. 0 = PCM 1 = Non-PCM
16	SPDIF PB Reg Validity	RW 0x0	Validity bit value when <SPDIF PB Enable Mem Validity> = 0. **NOTE:** During Non-PCM stream, this bit should be set to 1, to comply with the SPDIF standard.
15:10	RSVD	RW 0x0	Always write 0 to this field.
9:6	Reserved	RSVD 0x0	Reserved
5	RSVD	RW 0x0	Always write 0 to this bit.
4	Force Parity Error	RW 0x0	When this bit is set, if the Perr bit (parity error indication bit) in memory is also asserted, then an erroneous parity will be sent. 0 = Disable: Force parity error from memory is disabled. 1 = Enable: Force parity error from memory is enabled.
3	RSVD	RW 0x0	Always write 0 to this field.
2	SPDIF PB Mem User En	RW 0x0	0 = User Buffered: User bit is used from the internal user buffers. 1 = User in Mem: User bit is read from memory.
1	SPDIF PB Enable Mem Validity	RW 0x0	0 = Register Validity: SPDIF Frame validity bit is set to <SPDIF PB Reg Validity>. 1 = Memory Validity: SPDIF Frame validity bit is set to the value that was read from memory.
0	SPDIF PB Block Start	RW 0x0	When both SPDIF user and status bits are taken from the Audio unit register file internal buffers, then this bit should be 0. Otherwise, configure it to 1. 0 = Internally: SPDIF frame block start is managed internally by the Audio unit. 1 = FIFO data word: SPDIF frame block start is controlled by the values present in the memory data word.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1315: SPDIF Playback Channel Status Left n Register (n=0–5)

Offset: status0: 0x000EA280, status1: 0x000EA284, status2: 0x000EA288, status3: 0x000EA28C, status4: 0x000EA290, status5: 0x000EA294

Offset Formula: 0x000EA280+n*4: where n (0-5) represents status

Bit	Field	Type/InitVal	Description
31:0	Left Channel Status Buffer	RW 0x0	Left Channel Status internal buffer. Register offset 0x80 represents bits [31:0] of left channel status bits. Register offset 0x94 represents bits [191:160] of the left channel status bits.

Table 1316: SPDIF Playback Channel Status Right n Register (n=0–5)

Offset: status0: 0x000EA2A0, status1: 0x000EA2A4, status2: 0x000EA2A8, status3: 0x000EA2AC, status4: 0x000EA2B0, status5: 0x000EA2B4

Offset Formula: 0x000EA2A0+n*4: where n (0-5) represents status

Bit	Field	Type/InitVal	Description
31:0	Right Channel Status Buffer	RW 0x0	Right Channel Status internal buffer. Register offset 0xA0 represents bits [31:0] of the right channel status bits. Register offset 0xB4 represents bits [191:160] of right channel status bits.

Table 1317: SPDIF Playback User Bits Left n Register (n=0–5)

Offset: User0: 0x000EA2C0, User1: 0x000EA2C4, User2: 0x000EA2C8, User3: 0x000EA2CC, User4: 0x000EA2D0, User5: 0x000EA2D4

Offset Formula: 0x000EA2C0+n*4: where n (0-5) represents User

Bit	Field	Type/InitVal	Description
31:0	Left Channel User Buffer	RW 0x0	Left Channel User bits internal buffer. Register offset 0xc0 represents bits [31:0] of the left channel user bits. Register offset 0xd4 represents bits [191:160] of the left channel user bits.

Table 1318: SPDIF Playback User Bits Right n Register (n=0–5)

Offset: User0: 0x000EA2E0, User1: 0x000EA2E4, User2: 0x000EA2E8, User3: 0x000EA2EC, User4: 0x000EA2F0, User5: 0x000EA2F4

Offset Formula: 0x000EA2E0+n*4: where n (0-5) represents User

Bit	Field	Type/InitVal	Description
31:0	Right Channel User Buffer	RW 0x0	Right Channel User bits internal buffer. Register offset 0xe0 represents bits [31:0] of the right channel user bits. Register offset 0xf4 represents bits [191:160] of the right channel user bits.

Copyright © 2015 Marvell

Table 1319: I2S Playback Control Register
Offset: 0x000EA508

Bit	Field	Type/InitVal	Description
31:30	I2S PB sample size	RW 0x0	I2S playback real sample size. Redundant bits are sent as zeros (I2S always uses 32 bits per channel). **NOTE:** The same sample size must be configured in the <Playback Sample Size> of the Playback Control Register. 0 = 32bit 1 = 24bit 2 = 20bit 3 = 16bit: Use this configuration for both 16 bit and 16 bit compact mode memory structure configuration.
29:26	I2S PB Justification	RW 0x5	**NOTE:** 1.The same value should be configured to <I2S Recording Justification>. 2. Bit [26] determines the polarity of I2S LRCLK. 0 = I2S_Left_J: I2S Left Justified 5 = I2S 8 = I2S_Right_J: I2S Right Justified
25:24	RSVD	RW 0x0	The value of this field must be 2'b0.
23	RSVD	RW 0x0	Always write 0 to this bit. **NOTE:**
22:0	Reserved	RSVD 0x0	Reserved

A.15.5 Audio Recording

Table 1320: Recording Control Register
Offset: 0x000E9000

Bit	Field	Type/InitVal	Description
31:11	Reserved	RSVD 0x0	Reserved
10	I2S Recording Enable	RW 0x0	Used to enable/disable I2S recording. **NOTE:** 1. If <I2S Recording Enable> is set to 0 and I2S playback is also disabled, then the I2S clocks are disabled. 2. If the I2S clocks should remain active while I2S Recording DMA should be stopped, use <Recording Pause> and leave I2S recording enabled. 0 = I2S recording disabled 1 = I2S recording enabled

Copyright © 2015 Marvell

Table 1320: Recording Control Register (Continued)
Offset: 0x000E9000

Bit	Field	Type/InitVal	Description
9	Recording Pause	RW 0x0	When set, the DMA engine stops working after it completes the transfer of the current burst of audio data to memory. The audio data received after the DMA stops is discarded. **NOTE:** 1. Pause differs from mute since during mute the DMA recording operation is on going. 2. I2S Pause differs from I2S stop (I2S Recording Enable is set to 0), since pause does not disable the I2S recording clocks. 0 = Pause disabled 1 = Pause enabled
8	Recording Mute	RW 0x0	When this bit is set, the DMA recording continues but the recorded data is forced to zero (except for SPDIF control bits). In normal operation, this bit must be cleared to zero. Mute is different from Pause: during pause the DMA recording stops. 0 = Mute disabled 1 = Mute enabled
7	RSVD	RW 0x0	Always write 0 to this bit.
6:5	Recording DMA Burst Size	RW 0x2	Specifies the burst size of the DMA. 1 = 32 Bytes 2 = 128 Bytes 3 = Reserved
4	Recording Mono	RW 0x0	When set, only one of the channels is recorded. Data arriving from the other channel is discarded. The <Recorded Mono Channel> bit determines which channel is recorded. 0 = Stereo mode: Mono mode is off. 1 = Mono mode
3	Recorded Mono Channel	RW 0x0	**NOTE:** Start of block is required to be used from memory if user bits are taken from memory and are required to align to the start of block (or if status bits could be used from memory). Left channel data is discarded. 0 = Left Channel Mono: Left channel is used in Mono recording mode. Right channel data is discarded. 1 = Right Channel Mono: Right channel is used in Mono recording mode. Left channel data is discarded.

Copyright © 2015 Marvell

Table 1320: Recording Control Register (Continued)
Offset: 0x000E9000

Bit	Field	Type/InitVal	Description
2:0	Recording Sample Size	RW 0x1	This field determines the memory structure during recording. **NOTE:** In the case of I2S also configure: the <I2S Recording Sample Size> in the I2S Recording Control Register. 0 = 32Bits: I2S mode only. Reserved for SPDIF. Each 32-bit word in memory includes 32 bits of audio data. 1 = 24Bits: Each 32-bit word in memory includes 24 bits of audio data at bits [23:0]. If this mode is used for a sampling size of less than 24 bits then bit [23] holds the MS bit of the data. Redundant LS bits will be 0 in the case of I2S. 2 = 20Bits: Each 32-bit word in memory includes 20 bits of audio data at bits [19:0]. 3 = 16Bits Compact: Each 32-bit word in memory consists of two halfwords of 16 bits of audio data. Using this mode for SPDIF means that the user and status bits are stored in the internal buffers and cannot be stored with the data. 7 = 16Bits: Each 32-bit word in memory includes 16 bits of audio data at bits [15:0].

Table 1321: Recording Start Address Register
Offset: 0x000E9004

Bit	Field	Type/InitVal	Description
31:0	Recording Buffer Start Address	RW 0x0	Contains pointer to the starting location of the recording buffer in memory. When recording starts, data in this location is the first to be received. **NOTE:** 1. Base Address must be 8-Byte aligned. 2. It is highly recommended that the address be aligned to the <Recording DMA Burst Size> as configured in the Recording Control register. This will save 50% of the DMA accesses to memory.

Table 1322: Recording Buffer Size Register
Offset: 0x000E9008

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1322: Recording Buffer Size Register (Continued)
Offset: 0x000E9008

Bit	Field	Type/InitVal	Description
21:0	Recording Buffer Size	RW 0x0	Contains the size of the recording buffer in Word units (32 bit) minus one. If set to 0, recording buffer is 1 Word (4 bytes total). If set to 1, recording buffer is 2 Words (8 bytes total). . . . If set to 0x3FFFFF (maximum allowed value), recording buffer size is 4M words or 16 MBytes. **NOTE:** Recording Buffer Size must be an integer multiple of <Recording DMA Burst Size>, as determined in the Recording Control register. The buffer size should be greater than the burst size.

Table 1323: Recording Buffer Byte Counter Register
Offset: 0x000E900C

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:0	Recording Buf Byte Count	RO 0x0	Number of bytes transferred by the recording DMA. Modulates the recording buffer size. **NOTE:** This counter is reset when its value reaches the value of the <Recording Buffer Size> field in the Recording Buffer Size Register or when recording is re-enabled.

Table 1324: I2S Recording Control Register
Offset: 0x000EA408

Bit	Field	Type/InitVal	Description
NOTE: Only 32-bit access is allowed to the I2S recording registers.			
31:30	I2S Recording Sample Size	RW 0x0	I2S Recording Sample Size. Redundant bits are written as zeros to memory. **NOTE:** The same sample size must also be configured in the <Recording Sample Size> field at the Recording Control register. 0 = 32bit 1 = 24bit 2 = 20bit 3 = 16bit

Copyright © 2015 Marvell

Table 1324: I2S Recording Control Register (Continued)
Offset: 0x000EA408

Bit	Field	Type/InitVal	Description
29:26	I2S Recording Justification	RW 0x5	**NOTE:** 1. Bit [26] determines the polarity of I2S LRCLK. 0 = Left: I2S data is in the left channel when I2S LRCLK is high. 1 = Right: I2S data is in the right channel when I2S LRCLK is high 2. The same value should be configured to <I2S Playback Justification> when recording and playback are concurrent. 0 = I2S_Left_J: I2S Left Justified. 5 = I2S 8 = I2S_Right_J: I2S Right Justified.
25	RSVD	RW 0x0	Always write 0 to this bit.
24:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.16 Serial Peripheral Interface (SPI) Registers

The following table provides a summarized list of all registers that belong to the Serial Peripheral Interface (SPI), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1325: Summary Map Table for the Serial Peripheral Interface (SPI) Registers

Register Name	Offset	Table and Page
SPI		
SPI Control Register (N=0–1)	SPI_Controller0: 0x00010600, SPI_Controller1: 0x00010680	Table 1326, p. 1074
SPI Interface Configuration Register (N=0–1)	SPI_Controller0: 0x00010604, SPI_Controller1: 0x00010684	Table 1327, p. 1074
SPI Data Out Register (N=0–1)	SPI_Controller0: 0x00010608, SPI_Controller1: 0x00010688	Table 1328, p. 1076
SPI Data In Register (N=0–1)	SPI_Controller0: 0x0001060C, SPI_Controller1: 0x0001068C	Table 1329, p. 1076
SPI Interrupt Cause Register (N=0–1)	SPI_Controller0: 0x00010610, SPI_Controller1: 0x00010690	Table 1330, p. 1076
SPI Interrupt Mask Register (N=0–1)	SPI_Controller0: 0x00010614, SPI_Controller1: 0x00010694	Table 1331, p. 1076
SPI Timing Parameters 1 Register (N=0–1)	SPI_Controller0: 0x00010618, SPI_Controller1: 0x00010698	Table 1332, p. 1077
SPI Timing Parameters 2 Register (N=0–1)	SPI_Controller0: 0x0001061C, SPI_Controller1: 0x0001069C	Table 1333, p. 1078
SPI Direct Write Configuration Register (N=0–1)	SPI_Controller0: 0x00010620, SPI_Controller1: 0x000106A0	Table 1334, p. 1078
SPI Direct Write Header Register (N=0–1)	SPI_Controller0: 0x00010624, SPI_Controller1: 0x000106A4	Table 1335, p. 1079
SPI Direct Read Header Register (N=0–1)	SPI_Controller0: 0x00010628, SPI_Controller1: 0x000106A8	Table 1336, p. 1079
SPI CS Address Decode Register (N=0–1)	SPI_Controller0: 0x0001062C, SPI_Controller1: 0x000106AC	Table 1337, p. 1079
SPI CSn Timing Parameters Register (n=0–7, N=0–1)	SPI_Controller0SPI CS0: 0x00010630, SPI_Controller0SPI CS1: 0x00010634...SPI_Controller1 SPI CS7: 0x000106CC	Table 1338, p. 1080
SPI Controller Version Register (N=0–1)	SPI_Controller0: 0x00010650, SPI_Controller1: 0x000106D0	Table 1339, p. 1081

Copyright © 2015 Marvell

A.16.1 SPI

Table 1326: SPI Control Register (N=0–1)
Offset: SPI_Controller0: 0x00010600, SPI_Controller1: 0x00010680
Offset Formula: 0x00010600+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved
4:2	SPI CS num	RW 0x0	This field defines which SPI_CS will be asserted when <CSnAct>=1. It has a binary format, thus, at any given point of time at most one SPI_CS will be active. 0 = SPI_CSn[0] 1 = SPI_CSn[1] 2 = SPI_CSn[2] 3 = SPI_CSn[3]
1	SMemRdy	RO 0x1	Serial Memory Data Transfer Ready
0	CSnAct	RW 0x0	Serial Memory Control 0 = Deactivated: Serial Memory Interface is deactivated. The SPI_CSn output is driven HIGH. 1 = Activated: Serial Memory Interface is activated. The SPI_CSn output is driven LOW.

Table 1327: SPI Interface Configuration Register (N=0–1)
Offset: SPI_Controller0: 0x00010604, SPI_Controller1: 0x00010684
Offset Formula: 0x00010604+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	Reserved	RW 0x0	Reserved Must be 0.
14	RxLSBF	RW 0x0	Receive LSB-First Enable 1 = Data Received is treated as least significant bit first. 0 = Data Received is treated as most significant bit first.
13	TxLSBF	RW 0x0	Transmit LSB-First Enable 1 = Enable; Data is transferred least significant bit first. 0 = Disable; Data is transferred most significant bit first.
12	CPHA	RW 0x0	SPI Clock Phase Bit This bit is used to shift the SCK serial clock. 0 = Half Cycle In: The first SCK edge is issued one-half cycle into the transfer operation. 1 = Beginning: The first SCK edge is issued at the beginning of the transfer operation.

Copyright © 2015 Marvell

Table 1327: SPI Interface Configuration Register (N=0–1) (Continued)
Offset: SPI_Controller0: 0x00010604, SPI_Controller1: 0x00010684
Offset Formula: 0x00010604+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
11	CPOL	RW 0x0	SPI Clock Polarity Bit This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. 0 = Active High: Active-high clocks selected. SCK idles low. 1 = Active Low: Active-low clocks selected. SCK idles high.
10	DirectRdHighSpeed	RW 0x0	SPI Direct Read High Speed Transaction Defines whether the SPI issues a dummy byte between the last byte of the address phase and the first byte of the data phase. For example, to activate an SPI Flash Fast_Read Command, the user should set this bit and configure <DirectRdHeader> to be 0x0B. 0 = Normal: Data will be driven on the byte following the last address byte. 1 = High Speed (HS): A dummy byte will be issued between the last Address byte and first data byte.
9:8	DirectAddrLen	RW 0x2	Address Length of SPI. This field defines the number of bytes of the address phase in a direct accesses (READ or WRITE) to SPI. For direct Writes, this field is considered only if <Direct_Wr_Addr_En> is active. 0 = 1Byte: SPI has a 1-byte address length. 1 = 2Byte: SPI has a 2-byte address length. 2 = 3Byte: SPI has a 3-byte address length. 3 = 4Byte: SPI has a 4-byte address length.
7:6	SPI_SPPR_HI	RW 0x0	SPI Baud Rate Pre-selection Bits 2,1 This field together with SPI_SPPR0 comprise the SPPR field. See SPR field description for a definition of the baud rate calculation.
5	BYTE_LEN	RW 0x0	Number of bytes in each serial flash I/O transfer. 0 = 1Byte 1 = 2Byte
4	SPI_SPPR0	RW 0x1	SPI Baud Rate Pre-selection Bit0 This field together with SPI_SPPR_HI comprise the SPPR field. See SPR field description for a definition of the baud rate calculation.
3:0	SPI_SPR	RW 0xd	SPI Baud Rate Selection Bits The SPR together with the SPPR define the SPI CLK frequency as follows: SPI actual frequency = core_clk / (SPR * (2 ^ SPPR)) Notes: 1. SPR and SPPR field must not be configured so that SPI freq = core_clk freq. 2. For odd division values, SPI_SCK will not be 50% duty-cycle.

Copyright © 2015 Marvell

Table 1328: SPI Data Out Register (N=0–1)
Offset: SPI_Controller0: 0x00010608, SPI_Controller1: 0x00010688
Offset Formula: 0x00010608+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	SMemOut	RW 0x0	Writing to this register initiates a transfer out to external serial memory. The content of this register will be shifted out serially. When BYTE_LEN = 1, bits [7:0] are shifted out. When BYTE_LEN=2, bits [7:0] are shifted out, followed by bits [15:8]. To shift in data from the serial memory, a dummy write can be used to advance SPI_SCK.

Table 1329: SPI Data In Register (N=0–1)
Offset: SPI_Controller0: 0x0001060C, SPI_Controller1: 0x0001068C
Offset Formula: 0x0001060C+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	SMemIn	RO 0x0	While SPI Data Out register bits are shifted out on SPI_MOSI, the value sampled on SPI_MISO is shifted into this register. When BYTE_LEN = 1, data is shifted into bit [0] and left shifted. When BYTE_LEN = 2, data is shifted into bits[7:0] first and then bits [15:8].

Table 1330: SPI Interrupt Cause Register (N=0–1)
Offset: SPI_Controller0: 0x00010610, SPI_Controller1: 0x00010690
Offset Formula: 0x00010610+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
NOTE: Write 0 to clear interrupt bits. Writing 1 does not effect the interrupt bits.			
31:1	Reserved	RSVD 0x0	Reserved
0	SMemRdiInt	RW0C 0x0	Serial Memory Data Transfer Ready Interrupt

Table 1331: SPI Interrupt Mask Register (N=0–1)
Offset: SPI_Controller0: 0x00010614, SPI_Controller1: 0x00010694
Offset Formula: 0x00010614+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1331: SPI Interrupt Mask Register (N=0–1) (Continued)
Offset: SPI_Controller0: 0x00010614, SPI_Controller1: 0x00010694
Offset Formula: 0x00010614+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
0	SMemRdiIntMask	RW 0x1	Mask Bit for the Serial Memory Data Transfer Ready Interrupt Mask only affects the assertion of the interrupt pin. It does not affect the setting of bits in the Cause register. 0 = Masked: Interrupt masked 1 = Enabled: Interrupt enabled

Table 1332: SPI Timing Parameters 1 Register (N=0–1)
Offset: SPI_Controller0: 0x00010618, SPI_Controller1: 0x00010698
Offset Formula: 0x00010618+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	SPI_TCS_HOLD	RW 0x4	SPI Timing CS Hold This field defines the minimal gap between last SPI_CLK edge and CS de-assertion. The gap is calculated in core_clk cycles.
11:8	SPI_TCS_SETUP	RW 0x4	SPI Timing CS Setup This field defines the minimal gap between CS assertion and first SPI_CLK edge. The gap is calculated in core_clk cycles.
7:6	SPI_TMISO_SAMPLE	RW 0x1	This field defines the gap between SPI CLK sampling edge (as determined by CPOL and CPHA) and the actual core clock cycle that MISO is captured. 0x0 = MISO sample occurs upon SPI_CLK edge. 0x1 = MISO sample occurs 1 core_clk after SPI_CLK edge. 0x2 = MISO sample occurs 2 core_clk after SPI_CLK edge. 0x3 = MISO sample occurs 3 core_clk after SPI_CLK edge. The user must set this field so that the sample does not occur before the following SPI_CLK edge. This means that SPI_CLK frequency as defined by <SPR> and <SPPR> and this field need to maintain: 1/2 SPI_CLK > TMISO_SAMPLE.
5:0	SP_TCSH	RW 0x3f	SPI Timing CS High Minimum Gap This field , together with <SPI_TCSH_9_6>, defines the minimal gap between CS de-assertion and CS assertion. The gap is calculated in number of core_clk cycles. **NOTE:** The initial value is maximal to enable a wide variety of Boot flash devices. To increase performance, Marvell recommends adding to the Boot code a configuration write that will change this field to be the minimal values suitable to the corresponding Boot device.

Table 1333: SPI Timing Parameters 2 Register (N=0–1)
Offset: SPI_Controller0: 0x0001061C, SPI_Controller1: 0x0001069C
Offset Formula: 0x0001061C+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0xF00	Reserved
3:0	SPI_TCSH_9_6	RW 0x0	SPI Timing CS High bits [9:6] This field, together with SPI_TCSH, defines the required gap between CS assertion and CS de-assertion. The actual gap is {SPI_TCSH_9_6, SPI_TCSH}.

Table 1334: SPI Direct Write Configuration Register (N=0–1)
Offset: SPI_Controller0: 0x00010620, SPI_Controller1: 0x000106A0
Offset Formula: 0x00010620+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:20	Reserved	RSVD 0x0	Reserved
19:17	Direct Wr Cs Hold	RW 0x0	In Direct Writes, this field defines the minimal number of core clock cycles between assertion of CS and the first SPI_CLK.
16	Direct Wr Deassert Cs	RW 0x1	When this bit is active (high), Direct Writes will assert SPI_CS prior to the data transmission and de-assert SPI_CS upon completion of the data transmission. If this feature is de-activated (set to 0), then prior to any direct write, SPI_CS assertion should be guaranteed by software.
15:9	Reserved	RSVD 0x0	Reserved
8	Direct Wr Addr Enable	RW 0x0	When this bit is active (high), every direct write access to SPI will be prefixed with the first <DirectAddrLen> bytes of the request address.
7:3	Reserved	RSVD 0x0	Reserved
2:1	Direct Wr Hdr Size	RW 0x0	This field specifies the number of bytes to be transmitted as the data prefix. 0 = 1 byte: Hdr[7:0] will be transmitted. 1 = 2 bytes: Hdr[15:0] will be transmitted. 2 = 3 bytes: Hdr[23:0] will be transmitted. 3 = 4 bytes: Hdr[31:0] will be transmitted.
0	Direct Write Hdr Enable	RW 0x0	When this bit is active (high), every direct write access to SPI will be prefixed with the first <Direct Wr Hdr Size> bytes of <DirectWrHdrData>.

Copyright © 2015 Marvell

Table 1335: SPI Direct Write Header Register (N=0–1)
Offset: SPI_Controller0: 0x00010624, SPI_Controller1: 0x000106A4
Offset Formula: 0x00010624+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:0	SpiDirectHdr	RW 0x0	Upon a direct access to SPI, the value in this field will be added as a header to the data. The number of bytes that will be added is defined according to <Direct_Wr_Hdr_Size>.

Table 1336: SPI Direct Read Header Register (N=0–1)
Offset: SPI_Controller0: 0x00010628, SPI_Controller1: 0x000106A8
Offset Formula: 0x00010628+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	DirectRdHeader	RW 0x3	SPI Direct Read Header Defines the SPI command byte that will be issued upon a Direct Read transaction. Default value is the standard SPI Read Command. User should align configuration of this field with <DirectRdHighSpeed> and <DirectAddrLen> for a correct correlation with the target's protocol.

Table 1337: SPI CS Address Decode Register (N=0–1)
Offset: SPI_Controller0: 0x0001062C, SPI_Controller1: 0x000106AC
Offset Formula: 0x0001062C+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:22	Reserved	RSVD 0x0	Reserved
21	SPIMaskUsedAddressBits	RW 0x0	Mask used address bits. If SPIDecodeMode = 0x1 or 0x2, some address bits are used to determine SPI CS. Setting this field to 0x1 will mask the used address bits to zeros. The exact address bits used are determined according to fields <SPI DecodeMode>, <SPI Window0IndexLSB> and <SPI Window1IndexLSB> fields in this register. 0 = NoMask: Do not mask any address bits. 1 = Mask: Mask to 0 address bits which were used for selecting CSs.
20:16	SPI Window1IndexLSB	RW 0x0	SPI Window1 Index LSB When <SPI DecodeMode> is configured to DoubleWindow, this field defines the LSB Index of the two address bits that determines the target CS (of CS[7:4]) When <SPI DecodeMode> is configured to AttrPerCS or to SingleWindow , the value in this field has no affect.

Table 1337: SPI CS Address Decode Register (N=0–1) (Continued)
Offset: SPI_Controller0: 0x0001062C, SPI_Controller1: 0x000106AC
Offset Formula: 0x0001062C+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
15:13	Reserved	RSVD 0x0	Reserved
12:8	SPI Window0IndexLSB	RW 0x0	SPI Window0 Index LSB When <SPI DecodeMode> is configured to SingleWindow, this field defines the LSB Index of the three address bits that determines the target CS (of CS[7:0]) When <SPI DecodeMode> is configured to DoubleWindow, this field defines the LSB Index of the two address bits that determines the target CS (of CS[3:0]) When <SPI DecodeMode> is configured to AttrPerCS the value in this field has no affect.
7:2	Reserved	RSVD 0x0	Reserved
1:0	SPI DecodeMode	RW 0x0	SPI CS Decode Mode Define the mapping mode between an SPI direct access and its target CS. 0 = AttrPerCs: Every CS has its dedicated ATTR. The target CS is the binary encoding of {Attr[0],Attr[7:6]} 1 = SingleWindow: The CS is defined according to three consecutive bits of the address. The LSB index of the 3 bits is defined by <SPI_Window0IndexLSB>. 3 = DoubleWindow: There are two SPI Windows. SPI Window0 is mapped to CS0-3 and SPI Window1 is mapped to CS4-7. The Window is determined by the ATTR as follows: ATTR[0]=0 : SPI Window0 ATTR[0]=1 : SPI Window1 For Window0, the target CS (of CS[3:0]) is defined according to two consecutive bits of the address. The LSB of these two bits is defined by <SPI_Window0IndexLSB> For Window1, the target CS (of CS[7:4]) is defined according to two consecutive bits of the address. The LSB of these two bits is defined by <SPI_Window1IndexLSB>

Table 1338: SPI CSn Timing Parameters Register (n=0–7, N=0–1)
Offset: SPI_Controller0SPI CS0: 0x00010630, SPI_Controller0SPI CS1: 0x00010634...SPI_Controller1SPI CS7: 0x000106CC
Offset Formula: 0x00010630+4*n + 128*N : where N (0-1) represents SPI_Controller, where n (0-7) represents SPI CS

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:20	SPI_CSn_TCS_HOLD	RW 0x0	SPI CSn Hold See description at <TCS_HOLD> in register <SPI Timing Paramters 1>
19:16	SPI_CSn_TCS_SETUP	RW 0x0	SPI CSn Setup See description at <TCS_SETUP> in register <SPI Timing Paramters 1>
15	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 1338: SPI CSn Timing Parameters Register (n=0–7, N=0–1) (Continued)
Offset: SPI_Controller0SPI CS0: 0x00010630, SPI_Controller0SPI CS1: 0x00010634...SPI_Controller1SPI CS7: 0x000106CC
Offset Formula: 0x00010630+4*n + 128*N : where N (0-1) represents SPI_Controller, where n (0-7) represents SPI CS

Bit	Field	Type/InitVal	Description
14:13	SPI_CSn_TMISO_SAMPLE	RW 0x0	SPI CSn Timing Miso Sample See description at <TMISO_SAMPLE> in register <SPI Timing Paramters 1>
12	SPI_CSn_CPHA	RW 0x0	SPI CSn Clock Phase See description at <CPHA> in register <SPI Interface Configuration>
11	SPI_CSn_CPOL	RW 0x0	SPI CSn Clock Polarity See description at <CPOL> in register <SPI Interface Configuration>
10	SPI_CSn_DirectRdHighSpeed	RW 0x0	SPI CSn Read Direct accesses high speed See description at <DirectRdHighSpeed> in register <SPI Interface Configuration>
9:8	SPI_CSn_DirectAddrLen	RW 0x0	SPI CSn Direct accesses Address length See description at <DirectAddrLen> in register <SPI Interface Configuration>
7:5	SPI_CSn_SPPR	RW 0x0	SPI CSn SPPR See description at <SPI_SPR> in register <SPI Interface Configuration>
4:1	SPI_CSn_SPR	RW 0x0	SPI CSn SPR See description at <SPI_SPR> in register <SPI Interface Configuration>
0	SpiParamMode	RW 0x0	SPI CSn parameter mode 0 = Global: Transaction to SPI CS%n act according to the global set of paramters which are located at: <SPI Interface Configurations>, <SPI Timing Parameters 1>, <SPI Timing Parameters 2> 1 = Local: Transaction to SPI CS%n act accoriding to its dedicated parameters

Table 1339: SPI Controller Version Register (N=0–1)
Offset: SPI_Controller0: 0x00010650, SPI_Controller1: 0x000106D0
Offset Formula: 0x00010650+128*N: where N (0-1) represents SPI_Controller

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:4	SpiMainVersion	RO 0x1	Spi controller Version - Main field
3:0	SpiSubVersion	RO 0x1	Spi Controller Sub Version

Copyright © 2015 Marvell

A.17 Secure Digital Input/Output (SDIO) 3.0 Registers

The following table provides a summarized list of all registers that belong to the Secure Digital Input/Output (SDIO) 3.0, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1340: Summary Map Table for the Secure Digital Input/Output (SDIO) 3.0 Registers

Register Name	Offset	Table and Page
Secure Digital Input/Output (SDIO)		
System Address Register	0x000D8000	Table 1341, p. 1083
Block Count and Size Register	0x000D8004	Table 1342, p. 1084
Command Argument Register	0x000D8008	Table 1343, p. 1084
Command and Transfer Mode Register	0x000D800C	Table 1344, p. 1085
Response 0 and 1 Register	0x000D8010	Table 1345, p. 1086
Response 2 and 3 Register	0x000D8014	Table 1346, p. 1087
Response 4 and 5 Register	0x000D8018	Table 1347, p. 1087
Response 6 and 7 Register	0x000D801C	Table 1348, p. 1087
Buffer Data Port 0 and 1 Register	0x000D8020	Table 1349, p. 1088
Present State Register	0x000D8024	Table 1350, p. 1088
Block Gap and Host Control Register	0x000D8028	Table 1351, p. 1090
Timeout Control/Software Reset and Clock Control Register	0x000D802C	Table 1352, p. 1093
Normal and Error Interrupt Status Register	0x000D8030	Table 1353, p. 1095
Normal and Error Interrupt Status Enable Register	0x000D8034	Table 1354, p. 1097
Normal and Error Interrupt Status Interrupt Enable Register	0x000D8038	Table 1355, p. 1099
Auto Host Control2 and Command12 Error Status Register	0x000D803C	Table 1356, p. 1102
Capabilities 1 and 0 Register	0x000D8040	Table 1357, p. 1103
Capabilities 2 and 3 Register	0x000D8044	Table 1358, p. 1105
Maximum Current 1 and 2 Register	0x000D8048	Table 1359, p. 1106
Maximum Current 3 and 4 Register	0x000D804C	Table 1360, p. 1107
Force Event Auto Command 12 Error Register	0x000D8050	Table 1361, p. 1107
ADMA Error Status Register	0x000D8054	Table 1362, p. 1108
ADMA System Address Register High	0x000D8058	Table 1363, p. 1109
ADMA System Address Register Low	0x000D805C	Table 1364, p. 1109
Slot Interrupt Status Register	0x000D80FC	Table 1365, p. 1109
SD Extra Parameters Register	0x000D8100	Table 1366, p. 1110
FIFO Parameters Register	0x000D8104	Table 1367, p. 1110
SPI Mode and Clock and Burst Size Setup Register	0x000D8108	Table 1368, p. 1112
CE-ATA Register	0x000D810C	Table 1369, p. 1113
Drive Strength Register	0x000D8110	Table 1370, p. 1114
RX Configuration Register	0x000D8114	Table 1371, p. 1115
TX Configuration Register	0x000D8118	Table 1372, p. 1115

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1340: Summary Map Table for the Secure Digital Input/Output (SDIO) 3.0 Registers (Continued)

Register Name	Offset	Table and Page
SD Host Controller Version 0 Register	0x000D81F8	Table 1373, p. 1116
SD Host Controller Version 1 Register	0x000D81FC	Table 1374, p. 1116
Mbus to AXI Bridge Control		
Mbus Port Control Register	0x000DC000	Table 1375, p. 1116
AXI Port Control Register	0x000DC004	Table 1376, p. 1117
AXI to Mbus Bridge Window and Control		
Bridge Window Control Register (N=0–7)	Windows0: 0x000DC080, Windows1: 0x000DC088, Windows2: 0x000DC090, Windows3: 0x000DC098, Windows4: 0x000DC0A0, Windows5: 0x000DC0A8, Windows6: 0x000DC0B0, Windows7: 0x000DC0B8	Table 1377, p. 1117
Bridge Window Base Register (N=0–7)	Windows0: 0x000DC084, Windows1: 0x000DC08C, Windows2: 0x000DC094, Windows3: 0x000DC09C, Windows4: 0x000DC0A4, Windows5: 0x000DC0AC, Windows6: 0x000DC0B4, Windows7: 0x000DC0BC	Table 1378, p. 1118
Mbus Control Register	0x000DC0C0	Table 1379, p. 1118
Sync Barrier WindowN Control Register (N=0–1)	SyncbWin0: 0x000DC0C4, SyncbWin1: 0x000DC0CC	Table 1380, p. 1119
Sync Barrier WindowN Base Register (N=0–1)	SyncbWin0: 0x000DC0C8, SyncbWin1: 0x000DC0D0	Table 1381, p. 1119

A.17.1 Secure Digital Input/Output (SDIO)

Table 1341: System Address Register
Offset: 0x000D8000

Bit	Field	Type/InitVal	Description
31:16	DMA_ADDR_H	RW 0x0	DMA Address High 16 MSb of DMA system buffer starting byte address. This field is also used with the Auto Cmd 23 to set a 32-bit block count value to the argument of cmd23. This register holds the upper 16 bits of the cmd23 argument.
15:0	DMA_ADDR_L	RW 0x0	DMA Address Low 16 LSb of DMA system buffer starting byte address. This field is also used with the Auto Cmd 23 to set a 32-bit block count value to the argument of cmd23. This register holds the lower 16 bits of the cmd23 argument.

Copyright © 2015 Marvell

Table 1342: Block Count and Size Register
Offset: 0x000D8004

Bit	Field	Type/InitVal	Description
31:16	BLOCK_COUNT	RW 0x0	Block Count The Host Driver sets this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. 0x1 = 1 block ... 0xFFFF = 65535 blocks.
15	Reserved	RSVD 0x0	Reserved
14:12	HOST_DMA_BDRY	RW 0x0	Host DMA Buffer Boundary These bits specify the size of contiguous buffer in the system memory. If this boundary is crossed, an interrupt (dma_int) is generated. This interrupt is reflected in the <Tx Ready> field in the Normal Interrupt Status Register. 0x0 = 4K bytes 0x1 = 8K bytes ... 0x7 = 512K bytes
11:0	BLOCK_SIZE	RW 0x0	Block Size Indicates the block size of transferred data.

Table 1343: Command Argument Register
Offset: 0x000D8008

Bit	Field	Type/InitVal	Description
31:16	ARG_H	RW 0x0	Argument High 16 MSb of Command Argument. This value is inserted into the 48 bits command token bits[39:24].
15:0	ARG_L	RW 0x0	Argument Low 16 LSb of Command Argument. This value is inserted into the 48 bits command token bits[23:8].

Copyright © 2015 Marvell

Table 1344: Command and Transfer Mode Register
Offset: 0x000D800C

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:24	CMD_INDEX	RW 0x0	Command Index These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format. These bits are inserted into Command token bits[45:40].
23:22	CMD_TYPE	RW 0x0	Command Type There are 3 types of special commands: Suspend, Resume and Abort. These bits should be set to 00b for all other commands. 0 = Normal command 1 = Suspend command 2 = Resume command 3 = Abort command
21	DATA_PRESENT	RW 0x0	Data Present This bit is set to 1 to indicate that data is present and shall be transferred, using the DAT line. It is set to 0 for the following: - Commands using only CMD line (for example, CMD52). - Commands with no data transfer but using busy signal on DAT[0] line (R1b or R5b, for example, CMD38) - Resume command 0 = No data present 1 = Data present
20	CMD_INDEX_CHK_EN	RW 0x0	Command Index Check Enable If this bit is set to 1, the Host Controller checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 0 = Disable 1 = Enable
19	CMD_CRC_CHK_EN	RW 0x0	Command CRC Check Enable If this bit is set to 1, the Host Controller checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The position of CRC field is determined according to the length of the response. 0 = Disable 1 = Enable
18	Reserved	RSVD 0x0	Reserved
17:16	RESP_TYPE	RW 0x0	Response Type Select for SD/SD in SPI Modes 0 = No response 1 = 136 bit response: Response length is 136 bits 2 = 48 bit response: Response length is 48 bits 3 = 48 bit response and busy: Response length is 48 bits and check busy after response.
15:6	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1344: Command and Transfer Mode Register (Continued)
Offset: 0x000D800C

Bit	Field	Type/InitVal	Description
5	MULTI_BLK_SEL	RW 0x0	Multiple Block Select This bit should be set to 1 only when multiple blocks are to be transferred.
4	TO_HOST_DIR	RW 0x0	Data Transfer Direction Select This bit defines the direction of the MMC1_DAT[3:0] line data transfer. This bit is set to 1 by the Host Driver to transfer data from the SD card to the SD host controller. It is set to 0 for all other commands.
3:2	AUTO_CMD_EN	RW 0x0	Auto CMD Enable This field determines the use of auto command functions. 0 = Auto Command disabled 1 = Auto CMD12 enable 2 = Auto CMD23 enable 3 = Reserved
1	BLK_CNT_DIS	RW 0x0	Block Count Disable This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the Block Count register is disabled, which is useful in executing an infinite transfer. 0 = Enable 1 = Disable
0	DMA_EN	RW 0x0	DMA Enable This bit enables DMA functionality. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation begins when the Host Driver writes to the upper byte of Command register. If Programmed Input/Output (PIO) mode is required, this bit should be reset to 0.

Table 1345: Response 0 and 1 Register
Offset: 0x000D8010

Bit	Field	Type/InitVal	Description
31:16	RESP1	RO 0x0	Response 1 Contains bits[39:24] of the response token.
15:0	RESP0	RO 0x0	Response 0 Contains bits[23:8] of the response token.

Copyright © 2015 Marvell

Table 1346: Response 2 and 3 Register
Offset: 0x000D8014

Bit	Field	Type/InitVal	Description
31:16	RESP3	RO 0x0	Response 3 For 48-bits response token, this register is don't care. For 136-bits response token, this register contains bits[71:56] of the response token.
15:0	RESP2	RO 0x0	Response 2 For 48-bits response token, this register is don't care. For 136-bits response token, this register contains bits[55:40] of the response token.

Table 1347: Response 4 and 5 Register
Offset: 0x000D8018

Bit	Field	Type/InitVal	Description
31:16	RESP5	RO 0x0	Response 5 For 48 bits response token, this register is don't care. For 136 bits response token, this register contains bits[103:88] of response token.
15:0	RESP4	RO 0x0	Response 4 For 48-bits response token, this register is don't care. For 136-bits response token, this register contains bits[87:72] of response token.

Table 1348: Response 6 and 7 Register
Offset: 0x000D801C

Bit	Field	Type/InitVal	Description
31:16	RESP7	RO 0x0	Response 7 For 48-bits response token, this register is don't care. For 136-bits response token, this register contains bits[127:120] of the response token. For Auto CMD12 response, this register contains bits[39:24] of the response token.
15:0	RESP6	RO 0x0	Response 6 For 48-bits response token, this register is don't care. For 136-bits response token, this register contains bits[119:104] of the response token. For Auto CMD12 response, this register contains bits[23:8] of the response token.

Copyright © 2015 Marvell

Table 1349: Buffer Data Port 0 and 1 Register
Offset: 0x000D8020

Bit	Field	Type/InitVal	Description
31:16	CPU_DATA1	RW 0x0	Processor Data 1 16 MSb of the buffer
15:0	CPU_DATA0	RW 0x0	Processor Data 0 16 LSb of the buffer

Table 1350: Present State Register
Offset: 0x000D8024

Bit	Field	Type/InitVal	Description
31	UHSII_IF_DET	RO 0x0	UHSII I/F Detect
30	UHSII_LANE_SYNC	RO 0x0	UHSII Lane Sync
29	UHSII_DORM_STATE	RO 0x0	UHSII Dormant State
28:25	Reserved	RSVD 0x0	Reserved
24	CMD_LEVEL	RO 0x1	MMC1_CMD Line Signal Level This status is used to check the MMC1_CMD line level to recover from errors and for debugging.
23:20	DAT_LEVEL_LSB	RO 0x0	MMC1_DAT[3:0] Line Signal Level This status is used to check the MMC1_DAT[3:0] line level to recover from errors and for debugging. This is especially useful in detecting the busy signal level from MMC1_DAT[0].
19	WRITE_PROT	RO 0x0	Write Protect This field reflects the position of the write_protect latch on the SD card. This field should be ignored if there is no such feature being provided by the card in use. 0 = Write protected 1 = Write enabled
18	CARD_DET	RO 0x1	Card Detect Level This field reflects the value of the MMC1_CD pin. This field is only used for testing. 0 = Card not detected. 1 = Card detected.
17	CARD_STABLE	RO 0x0	Card Stable This field is only used for testing. It indicates the debounced value of the card present condition. 0 = Card unstable 1 = Card stable
16	CARD_INSERTED	RO 0x0	Card Inserted This field indicates the presence of an SD card. 0 = Card not inserted 1 = Card inserted

Copyright © 2015 Marvell

Table 1350: Present State Register (Continued)
Offset: 0x000D8024

Bit	Field	Type/InitVal	Description
15:12	Reserved	RSVD 0x0	Reserved
11	BUFFER_RD_EN	RO 0x0	Buffer Read Enable This field changes from 0x0 to 0x1 when block data is ready in the buffer and from 0x1 to 0x0 when all the block data is read from the buffer.
10	BUFFER_WR_EN	RO 0x0	Buffer Write Enable This field changes from 0x0 to 0x1 when block data can be written to the buffer. If this bit is set to 0x1, the entire block can be written to the buffer. This field changes from 0x1 to 0x0 when all block data is written.
9	RX_ACTIVE	RO 0x0	Rx Active This field indicates that read transfer is active. This bit is set to 1 for either of the following conditions: - After the end bit of the read command. - When read operation is restarted by writing a 1 to the <Continue Request> field in the Block Gap Control register. This bit is cleared to 0 for any of the following conditions: - When the last data block as specified by block length is transferred to the system. - In case of ADMA2, end of read operation is designated by Descriptor Table. - When all valid data blocks in the Host Controller have been transferred to the system and no current block transfers are being sent as a result of the Stop At Block Gap Request being set to 1. 0 = Transferring data 1 = No valid data
8	TX_ACTIVE	RO 0x1	Tx Active This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the Host Controller. This bit is set in either of the following cases: - After the end bit of the write command. - When write operation is restarted by writing a 1 to the <Continue Request> field in the Block Gap Control register. This bit is cleared in either of the following cases: - After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) In case of ADMA2, transfer count is designated by Descriptor Table. - After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request 0 = Transferring data 1 = No valid data
7:4	DAT_LEVEL_MSB	RO 0x0	MMC1_DAT[7:4] Line Signal Level This status is used to check the MMC1_DAT[7:4] line level to recover from errors and for debugging.
3	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1350: Present State Register (Continued)
Offset: 0x000D8024

Bit	Field	Type/InitVal	Description
2	DAT_ACTIVE	RO 0x0	Data Line Active Defines the status of the data line. 0 = Data line is free. 1 = Data line is in use.
1	CMD_INHIBIT_DAT	RO 0x0	Command Inhibit Data Defines the host driver status for issuing data commands. 0 = Issued: Data command can be issued. 1 = Not issued: Data command cannot be issued.
0	CMD_INHIBIT_CMD	RO 0x0	Command Inhibit Command When cleared to 0, the MMC1_CMD line is not in use, and the Host controller can issue a command using the MMC1_CMD line. This bit is set after the command register is written. This bit is cleared when the command response is received. Even if the <Command Inhibit Data> field is set to 1, commands using only the MMC1_CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a command complete interrupt in the Normal Interrupt Status Register. If the Host controller cannot issue the command because of a command conflict error, this bit remains 1, and the command complete is not set.

Table 1351: Block Gap and Host Control Register
Offset: 0x000D8028

Bit	Field	Type/InitVal	Description
31:27	Reserved	RSVD 0x0	Reserved
26	W_REMOVAL	RW 0x0	Wake-up on Card Removal 0 = No wake-up event 1 = Enable wake-up event: Enables wake-up event on card removal detection.
25	W_INSERTION	RW 0x0	Wake-up on Card Insertion 0 = No wake-up event 1 = Enable wake-up event: Enables wake-up event on card insertion detection.
24	W_CARD_INT	RW 0x0	Wake-up on Card Interrupt 0 = No wake-up event 1 = Enable wake-up event: Enables wake-up event on card interrupt detection.
23:20	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1351: Block Gap and Host Control Register (Continued)
Offset: 0x000D8028

Bit	Field	Type/InitVal	Description
19	INT_BLK_GAP	RW 0x0	Block Gap Interrupt This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. 0 = Disable 1 = Enable
18	RD_WAIT_CTL	RW 0x0	Read Wait Control If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the MMC1_DAT[2] line by Host hardware. Otherwise, the Host controller has to stop the SD clock to hold read data. When the Host driver This field is looked at only at block gap. Within a block, hardware stalls the clock top stop read data if the host cannot accept any more data, for example, because FIFO is full. When this field is cleared by software, the operation continues. During read wait, software can issue a different command for a different operation as long as it does not require MMC1_DAT[3:0] lines. When it wants to continue the waiting operation, software needs t
17	CONT_REQ	RW 0x0	Continue Request This field restarts a transaction that was stopped using <Stop At Block Gap Request>. To cancel the stop at the block gap, set the <Stop At Block Gap Request> field to 0 and set this field to 1 to restart the transfer. The host controller automatically clears this field in the following cases: - In the case of a read transaction, the MMC1_DAT[3:0] Line Active changes from 0 to 1 as a read transaction restarts. - In case of a write transaction, the Write Transfer Active changes from 0 to 1 as the write transaction restarts. Therefore, it is not necessary for the Host driver to set this bit to 0. If <Stop At Block Gap Request> is set to 1, any write to this bit is ignored.
16	STOP_AT_BLOCK_GAP_REQ	RW 0x0	Stop At Block Gap Request This field is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the Host driver leaves this bit set to 1. Clearing both this field and the <Continue Request> field does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The Host controller stops the clock at block gap request for write transfer, but for read transfer, it stops the clock if <Read Wait Control> is 0. Otherwise, the host controller issues a Read Wait command to stop read data.
15:12	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1351: Block Gap and Host Control Register (Continued)
Offset: 0x000D8028

Bit	Field	Type/InitVal	Description
11:9	SD_BUS_VLT	RW 0x0	SD Bus Voltage By setting these bits, the Host Driver selects the voltage level for the SD/MMC card. 0x0 to 0x4 = Reserved 5 = 1.8V 6 = 3.0V 7 = 3.3V
8	SD_BUS_POWER	RW 0x0	SD Bus Power This field controls the power going out to the SD card. It is cleared if one of the following occurs: - The sd_bus_vlt and the voltage support in the Capabilities Register 1 do not match. OR - If a card removal state was detected. The <SD Bus Voltage> field and this field should be configured to correct values. These fields do not actually do the voltage selection or switch power to the SD card. However, these fields are used internally to gate the clock.
7	CARD_DET_S	RW 0x0	Card Detect Signal Selection This field selects the source for card detection. When the source for card detection is switched, the interrupt should be disabled during the switching period by clearing the Normal Interrupt Status Enable Register to mask unexpected interrupts being caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing. 0 = Card detect input pin: (normal use) 1 = Card detect test level: (for debugging purposes only).
6	CARD_DET_L	RW 0x0	Card Detect Test Level This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not. 0 = No card inserted 1 = Card inserted
5	EX_DATA_WIDTH	RW 0x0	Extended Data Transfer Width This bit enables the 8-bit mode. 0 = Disable: Data width for bus mode is determined by <DATA_WIDTH>. 1 = Enable: 8-bit data width enabled
4:3	DMA_SEL	RW 0x0	DMA Select One of supported DMA modes as selected. The host driver checks support of DMA modes using the Capabilities Register 1. Use of the selected DMA is determined by the <DMA Enable> field in the Transfer Mode Register. 0 = SDMA 1 = ADMA 1 2 = 32-bit address ADMA2 3 = Reserved

Copyright © 2015 Marvell

Table 1351: Block Gap and Host Control Register (Continued)
Offset: 0x000D8028

Bit	Field	Type/InitVal	Description
2	HI_SPEED_EN	RW 0x0	Extend Data Output Enable This bit is optional. Before setting this bit, the Host Driver checks the High Speed Support in the Capabilities register. If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25 MHz). If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock. The eMMC/SDIO command and data lines are always driven from the rising edge of the clock. 1 = High speed mode 0 = Normal speed mode
1	DATA_WIDTH	RW 0x0	Data Width This bit selects the data width of the Host Controller. The Host Driver sets it to match the data width of the SD card. 0 = 1-bit data mode: using only MMC1_DAT[0]. This bit configuration is valid only when <EX_DATA_WIDTH>=0x0. 1 = 4-bit data mode
0	LED_CTRL	RW 0x0	LED Control This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction. 0 = LED off 1 = LED on

Table 1352: Timeout Control/Software Reset and Clock Control Register
Offset: 0x000D802C

Bit	Field	Type/InitVal	Description
31:27	Reserved	RSVD 0x0	Reserved
26	SW_RST_DAT	RW 0x0	Soft Reset for Data Port of Logic Only part of data circuit is reset. DMA circuit is also reset. 0 = Reset 1 = Normal
25	SW_RST_CMD	RW 0x0	Soft Reset for Command Part of Logic Only part of command circuit is reset. 0 = Normal 1 = Reset
24	SW_RST_ALL	RW 0x0	Software Reset for All This reset affects the entire Host Controller except for the card detection circuit. This reset affects the status, state machine, and FIFOs synchronously. This field also resets all private registers.

Table 1352: Timeout Control/Software Reset and Clock Control Register (Continued)
Offset: 0x000D802C

Bit	Field	Type/InitVal	Description
23:20	Reserved	RSVD 0x0	Reserved
19:16	TIMEOUT_VALUE	RW 0x0	Timeout Value Determines the interval by which MMC1_DAT[3:0] line timeouts are detected. This timeout is initiated in the following cases: - For a read transaction, waiting for data from cards. This is referred to as N<sub AC> timing value in the SD specification, which specifies the maximum timing from read command to read data (card data access time). - For a write transaction, waiting for data from the AXI slave, AXI Master, or processor, or waiting for the CRC status of the write block. 0x0 = SDCLK x 2^13 0x1 = SDCLK x 2^14 ... 0xE = SDCLK x 2^27 For example, if sd_clk_ frequency = base value, which is 200 MHz, then timeout base = 50 MHz (period = 20 ns). Then for 0xe setting, timeout value = 2^27 * 20ns =~ 2.684 seconds. For other transactions, there are fixed timeouts defined as follows (units in SDCLK cycles): On the card: N<sub CR> = 64, maximum timing value from command to response - N<sub ID> = 64 (5 in specification), maximum timing value from command to OCR.
15:8	SD_FREQ_SEL_LO	RW 0x0	SDCLK Frequency Select Lower Bits This, along with <SD_FREQ_SEL_HI>, define the clock divider value to be used by the host controller. The final SD_FREQ_SEL = {SD_FREQ_SEL_HI[1:0],SD_FREQ_SEL_LO[7:0]}. The selected value is multiplied by 2 to get the actual value. For example: The clock frequency of the base clock for this field and <SD_FREQ_SEL_HI> is defined by the field <BASE_FREQ> in Capabilities Register 1.
7:6	SD_FREQ_SEL_HI	RW 0x0	SDCLK Frequency Select Higher Bits This, along with <SD_FREQ_SEL_LO>, define the clock divider value to be used by the host controller. The final SD_FREQ_SEL = {SD_FREQ_SEL_HI[1:0],SD_FREQ_SEL_LO[7:0]}. The selected value is multiplied by 2 to get the actual divide value. For example: SD_FREQ_SEL = 0x00 = Base clock SD_FREQ_SEL = 0x01 = Divide by 2 of base clock SD_FREQ_SEL = 0x02 = Divide by 4 of base clock SD_FREQ_SEL = 0x3 = Divide by 6 of base clock. ... SD_FREQ_SEL = 0x3FF = Divide by 2046 of base clock. The clock frequency of the base clock for this field and <SD_FREQ_SEL_LO> is defined by the field <BASE_FREQ> in Capabilities Register 1.

Copyright © 2015 Marvell

Table 1352: Timeout Control/Software Reset and Clock Control Register (Continued)
Offset: 0x000D802C

Bit	Field	Type/InitVal	Description
5	CLK_GEN_SEL	RW 0x0	Clock Generator Select This field selects the clock generator mode. 0 = Divided Clock mode 1 = Programmable Clock mode
4:3	Reserved	RSVD 0x0	Reserved
2	SD_CLK_EN	RW 0x0	SDCLK Clock Enable This bit controls the SDCLK to the card. Before using the card, this bit should be set during the initialization phase.
1	INT_CLK_STABLE	RO 0x0	Internal Clock Stable This field is set to 1 once the controller detects that the internal clock is stable after setting of the <Internal Clock Enable> field.
0	INT_CLK_EN	RW 0x0	Internal Clock Enable This field controls the SDCLK to the internal logic. 0 = Disable 1 = Enable

Table 1353: Normal and Error Interrupt Status Register
Offset: 0x000D8030

Bit	Field	Type/InitVal	Description
31	CRC_STATUS_ERR	RW 0x0	CRC Status Error Set when the CRC status start bit or CRC status end bit or boot ACK status, returned from the card in write transaction has errors.
30	CPL_TIMEOUT_ERR	RW 0x0	Command Completion Signal Timeout Error This field is applicable for CE-ATA mode only. Set when a command completion signal timeout occurred.
29	AXI_RESP_ERR	RW 0x0	AXI Bus Response Error Set when a response other than "OKAY" was received on the AXI bus.
28	SPI_ERR	RW 0x0	SPI Mode Error 0 = No error occurred. 1 = Error occurred in SPI mode.: The cause can be determined by reading the <SPI Error Token> field in the SPI Mode Register.
27:26	Reserved	RSVD 0x0	Reserved
25	ADMA_ERR	RW 0x0	ADMA (Advanced Direct Memory Access) Error This bit is set when the host controller detects any errors during an ADMA-based data transfer. The ADMA state at the time an error occurs is saved in the ADMA Error Status Register. The host controller also generates this interrupt when it detects any invalid descriptor data. The <ADMA Error State> field in the ADMA Error Status Register indicates the state in which an error occurred. The host driver may find that a Valid bit is not set at the error descriptor. 0 = No error 1 = Error

Copyright © 2015 Marvell

Table 1353: Normal and Error Interrupt Status Register (Continued)
Offset: 0x000D8030

Bit	Field	Type/InitVal	Description
24	AUTO_CMD12_ERR	RW 0x0	Auto CMD12 Error Occurs when detecting that one of the bits in Auto CMD12 Error Status Register has changed from 0 to 1.
23	CUR_LIMIT_ERR	RW 0x0	Current Limit Error This feature is not supported and this bit is always read as 0.
22	RD_DATA_END_BIT_ERR	RW 0x0	Read Data End Bit Error Set when 0 was detected at the end bit position of read data that uses the MMC1_DAT[3:0] line or at the end bit position of the CRC status.
21	RD_DATA_CRC_ERR	RW 0x0	Read Data CRC Error Set when Read data that uses the MMC1_DAT[3:0] line transferred or Write CRC status having a value other than 010 was detected.
20	DATA_TIMEOUT_ERR	RW 0x0	Data Timeout Error Set when one of the following is detected: - Busy timeout after write CRC status - Write CRC status timeout - Read data timeout
19	CMD_INDEX_ERR	RW 0x0	Command Index Error 0 = No error: No command index error has occurred in the command response. 1 = Error: Command index error has occurred in the command response.
18	CMD_END_BIT_ERR	RW 0x0	Command End Bit Error 0 = End bit 1: 0 = Detection of end bit of a command response is 1. 1 = End bit 0: Detection of end bit of a command response is 0.
17	CMD_CRC_ERR	RW 0x0	Command CRC Error Set in two cases: - A CRC error is detected in the command response. - The Host controller detects a MMC1_CMD line conflict by monitoring the MMC1_CMD line when a command is issued. The Host controller aborts the command (stops driving MMC1_CMD line). The <Command Timeout Error> field is also set to 1 to distinguish.
16	CMD_TIMEOUT_ERR	RW 0x0	Command Timeout Error Set when no response is returned within 64 SDCLK cycles from the end bit of the command.
15	ERR_INT	RO 0x0	Error Interrupt If any of bits in the Error Interrupt Status Register are set, then this bit is set.
14:9	Reserved	RSVD 0x0	Reserved
8	CARD_INT	RO 0x0	Card Interrupt Set when the host controller detects an interrupt from the card.
7	CARD_REM_INT	RW 0x0	Card Removal Interrupt Set when a card removal event is detected.
6	CARD_INS_INT	RW 0x0	Card Insertion Interrupt Set when a card insertion event is detected.
5	RX_RDY	RW 0x0	Rx Ready This status is set if the <Buffer Read Enable> field in the Present State Register 1 changes from 0x0 to 0x1.

Copyright © 2015 Marvell

Table 1353: Normal and Error Interrupt Status Register (Continued)
Offset: 0x000D8030

Bit	Field	Type/InitVal	Description
4	TX_RDY	RW 0x0	Tx Ready This status is set if the <Buffer Write Enable> field in the Present State Register 1 changes from 0x0 to 0x1.
3	DMA_INT	RW 0x0	DMA Interrupt This status is set if the Host Controller detects DMA crossing over the <Host DMA Buffer Boundary> field in the Block Size Register.
2	BLOCK_GAP_EVT	RW 0x0	Block Gap Event If the <Stop At Block Gap Request> field in the Block Gap and Host Control Register is set, this field is set when both a read/write transaction is stopped at a block gap. If the <Stop At Block Gap Request> field is not set to 1, this bit is not set.
1	XFER_COMPLETE	RW 0x0	Transfer Complete This bit is set when a read/write transaction is completed. For a read transaction, this bit is set at the falling edge of Read Transfer Active Status. There are two cases in which this occurs: - Data transfer is completed as specified by data length. - Data stopped at the block gap and completed data transfer by setting the <Stop At Block Gap Request> field in the SD Block Gap and Host Control Register field. For a write transaction, this bit is set at the falling edge of the MMC1_DAT[3:0] Line Active status. There are two cases in which this occurs: - Data transfer is completed as specified by data length and the busy signal is released. - Data stopped at the block gap and completed data transfer by setting the <Stop At Block Gap Request> field.
0	CMD_COMPLETE	RW 0x0	Command Complete This bit is set when the end bit of the command response (Except Auto CMD12) is received. **NOTE:** Command Timeout Error has higher priority than Command Complete.

Table 1354: Normal and Error Interrupt Status Enable Register
Offset: 0x000D8034

Bit	Field	Type/InitVal	Description
31	CRC_STATUS_ERR_EN	RW 0x0	CRC Status Error Enable 0 = Disable 1 = Enable
30	CPL_TIMEOUT_ERR_EN	RW 0x0	CPL Timeout Error Enable 0 = Disable 1 = Enable
29	AXI_RESP_ERR_EN	RW 0x0	AXI Response Error Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1354: Normal and Error Interrupt Status Enable Register (Continued)
Offset: 0x000D8034

Bit	Field	Type/InitVal	Description
28	SPI_ERR_EN	RW 0x0	SPI Error Enable 0 = Disable 1 = Enable
27:26	Reserved	RSVD 0x0	Reserved
25	ADMA_ERR_EN	RW 0x0	ADMA Error Enable 0 = Disable 1 = Enable
24	AUTO_CMD12_ERR_EN	RW 0x0	Auto CMD12 Error Enable 0 = Disable 1 = Enable
23	CUR_LIM_ERR_EN	RW 0x0	Current Limit Error Enable 0 = Disable 1 = Enable
22	RD_DATA_END_BIT_ERR_EN	RW 0x0	Data End Bit Error Enable 0 = Disable 1 = Enable
21	RD_DATA_CRC_ERR_EN	RW 0x0	Data CRC Error Enable 0 = Disable 1 = Enable
20	DATA_TIMEOUT_ERR_EN	RW 0x0	Data Timeout Error Enable 0 = Disable 1 = Enable
19	CMD_INDEX_ERR_EN	RW 0x0	Command Index Error Enable 0 = Disable 1 = Enable
18	CMD_END_BIT_ERR_EN	RW 0x0	Command End Bit Error Enable 0 = Disable 1 = Enable
17	CMD_CRC_ERR_EN	RW 0x0	Command CRC Error Enable 0 = Disable 1 = Enable
16	CMD_TIMEOUT_ERR_EN	RW 0x0	Command Timeout Error Enable 0 = Disable 1 = Enable
15:13	Reserved	RSVD 0x0	Reserved
12	RETUNE_INT_EN	RW 0x0	Re-tuning Interrupt Event Status Enable 0 = Disable 1 = Enable
11	INT_C_INT_EN	RW 0x0	INT_C Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1354: Normal and Error Interrupt Status Enable Register (Continued)
Offset: 0x000D8034

Bit	Field	Type/InitVal	Description
10	INT_B_INT_EN	RW 0x0	INT_B Enable 0 = Disable 1 = Enable
9	INT_A_INT_EN	RW 0x0	INT_A Enable 0 = Disable 1 = Enable
8	CARD_INT_EN	RW 0x0	Card Interrupt Enable 0 = Disable 1 = Enable
7	CARD_REM_EN	RW 0x0	Card Removal Status Enable 0 = Disable 1 = Enable
6	CARD_INS_EN	RW 0x0	Card Insertion Status Enable 0 = Disable 1 = Enable
5	RD_RDY_EN	RW 0x0	Buffer Read Ready Enable 0 = Disable 1 = Enable
4	TX_RDY_EN	RW 0x0	Buffer Write Ready Enable 0 = Disable 1 = Enable
3	DMA_INT_EN	RW 0x0	DMA Interrupt Enable 0 = Disable 1 = Enable
2	BLOCK_GAP_EVT_EN	RW 0x0	Block Gap Event Enable 0 = Disable 1 = Enable
1	XFER_COMPLETE_EN	RW 0x0	Transfer Complete Enable 0 = Disable 1 = Enable
0	CMD_COMPLETE_EN	RW 0x0	Command Complete Enable 0 = Disable 1 = Enable

Table 1355: Normal and Error Interrupt Status Interrupt Enable Register
Offset: 0x000D8038

Bit	Field	Type/InitVal	Description
31	CRC_STATUS_ERR_INT_EN	RW 0x0	CRC Status Error Interrupt Enable 0 = Disable 1 = Enable
30	CPL_TIMEOUT_ERR_INT_EN	RW 0x0	CPL Timeout Error Interrupt Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1355: Normal and Error Interrupt Status Interrupt Enable Register (Continued)
Offset: 0x000D8038

Bit	Field	Type/InitVal	Description
29	AXI_RESP_ERR_INT_EN	RW 0x0	AXI Response Error Interrupt Enable 0 = Disable 1 = Enable
28	SPI_ERR_INT_EN	RW 0x0	SPI Error Interrupt Enable 0 = Disable 1 = Enable
27:26	Reserved	RSVD 0x0	Reserved
25	ADMA_ERR_INT_EN	RW 0x0	ADMA Error Interrupt Enable 0 = Disable 1 = Enable
24	AUTO_CMD12_ERR_INT_EN	RW 0x0	Auto CMD12 Error Interrupt Enable 0 = Disable 1 = Enable
23	CUR_LIM_ERR_INT_EN	RW 0x0	Current Limit Error Interrupt Enable 0 = Disable 1 = Enable
22	RD_DATA_END_BIT_ERR_INT_EN	RW 0x0	Data End Bit Error Interrupt Enable 0 = Disable 1 = Enable
21	RD_DATA_CRC_ERR_INT_EN	RW 0x0	Data CRC Error Interrupt Enable 0 = Disable 1 = Enable
20	DATA_TIMEOUT_ERR_INT_EN	RW 0x0	Data Timeout Error Interrupt Enable 0 = Disable 1 = Enable
19	CMD_INDEX_ERR_INT_EN	RW 0x0	Command Index Error Interrupt Enable 0 = Disable 1 = Enable
18	CMD_END_BIT_ERR_INT_EN	RW 0x0	Command End Bit Interrupt Error Enable 0 = Disable 1 = Enable
17	CMD_CRC_ERR_INT_EN	RW 0x0	Command CRC Error Interrupt Enable 0 = Disable 1 = Enable
16	CMD_TIMEOUT_ERR_INT_EN	RW 0x0	Command Timeout Error Interrupt Enable 0 = Disable 1 = Enable
15:13	Reserved	RSVD 0x0	Reserved
12	RETUNE_INT_INT_EN	RW 0x0	Re-Tuning Interrupt Interrupt Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1355: Normal and Error Interrupt Status Interrupt Enable Register (Continued)
Offset: 0x000D8038

Bit	Field	Type/InitVal	Description
11	INT_C_INT_INT_EN	RW 0x0	INT_C Interrupt Interrupt Enable 0 = Disable 1 = Enable
10	INT_B_INT_INT_EN	RW 0x0	INT_B Interrupt Interrupt Enable 0 = Disable 1 = Enable
9	INT_A_INT_INT_EN	RW 0x0	INT_A Interrupt Interrupt Enable 0 = Disable 1 = Enable
8	CARD_INT_INT_EN	RW 0x0	Card Interrupt Interrupt Enable 0 = Disable 1 = Enable
7	CARD_REM_INT_EN	RW 0x0	Card Removal Interrupt Enable 0 = Disable 1 = Enable
6	CARD_INS_INT_EN	RW 0x0	Card Insertion Interrupt Enable 0 = Disable 1 = Enable
5	RX_RDY_INT_EN	RW 0x0	Buffer Read Ready Interrupt Enable 0 = Disable 1 = Enable
4	TX_RDY_INT_EN	RW 0x0	Buffer Write Ready Interrupt Enable 0 = Disable 1 = Enable
3	DMA_INT_INT_EN	RW 0x0	DMA Interrupt Interrupt Enable 0 = Disable 1 = Enable
2	BLOCK_GAP_EVT_INT_EN	RW 0x0	Block Gap Event Interrupt Enable 0 = Disable 1 = Enable
1	XFER_COMPLETE_INT_EN	RW 0x0	Transfer Complete Interrupt Enable 0 = Disable 1 = Enable
0	CMD_COMPLETE_INT_EN	RW 0x0	Command Complete Interrupt Enable 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1356: Auto Host Control2 and Command12 Error Status Register
Offset: 0x000D803C

Bit	Field	Type/InitVal	Description
31	PRE_VAL_EN	RW 0x0	Preset Value Enable 0 = Disable: SDCLK and Driver Strength are controlled by the host driver. 1 = Enable: Automatic selection by Preset Value are enabled.
30	ASYNC_INT_EN	RW 0x0	Asynchronous Interrupt Enable This bit can be set to 0x1 if a card supports asynchronous interrupts and SD_CAPABILITIES_1_0[async_int_support] is set to 0x1. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode.
29:24	Reserved	RSVD 0x0	Reserved
23	SAMPLING_CLK_SEL	RW 0x0	Sampling Clock Select The host controller uses this bit to select the sampling clock to receive CMD and DAT. This bit is set by the tuning procedure and is valid after the completion of tuning (when [exe_tuning] is cleared). 0 = Tuning is not complete. 1 = Tuning is complete.
22	EXE_TUNING	RW 0x0	Execute Tuning 0 = Aborts the tuning procedure. 1 = Starts the tuning procedure.: It is automatically cleared when the tuning procedure is complete. The result of tuning is indicated to [sampling_clk_sel].
21:20	DRV_STRENGTH_SEL	RW 0x0	Driver Strength Select Host Controller output driver in 1.8V signaling is selected by this field. In 3.3V signaling, this field is not effective. This field can be set depending on Driver Type A, C and D support bits in the Capabilities register.
19	SDH_V18_EN	RW 0x0	1.8V Signaling Enable 0 = Disable: 3.3V Signaling enabled. 1 = Enable: 1.8V Signaling enabled.
18:16	UHS_MODE_SEL	RW 0x0	UHS Mode Select Selects a UHS-I mode and is effective when <SDH_V18_EN> = 0x1. If <PRE_VAL_EN> field in this register is set to 0x1, the host controller sets SDCLK Frequency select, Clock generator select in the clock control. uhsii_mode = 3'h7 ddr50_mode = 3'h4 sdr104_mode = 3'h3 sdr50_mode = 3'h2 sdr25_mode = 3'h1
15:8	Reserved	RSVD 0x0	Reserved
7	CMD_NOT_ISSUED	RO 0x0	Command Not Issued Due to Auto_Cmd12 Error
6:5	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1356: Auto Host Control2 and Command12 Error Status Register (Continued)
Offset: 0x000D803C

Bit	Field	Type/InitVal	Description
4	AUTO_CMD_INDEX_ERR	RW 0x0	Auto CMD12 or Auto CMD23 Error This error occurs if the command index error occurs in response to a command. 0 = Disable 1 = Enable
3	AUTO_CMD_END_BIT_ERR	RW 0x0	Auto CMD12 or Auto CMD23End Bit Error When the end bit of a command response is 0, this error occurs. 0 = Disable 1 = Enable
2	AUTO_CMD_CRC_ERR	RW 0x0	Auto CMD12 or Auto CMD23 CRC Error When a CRC error in the command response is detected, this error occurs. 0 = Disable 1 = Enable
1	AUTO_CMD_TIMEOUT_ERR	RW 0x0	Auto CMD12 or Auto CMD23 Timeout Error This error occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. 0 = Disable 1 = Enable
0	AUTO_CMD12_NOT_EXE	RW 0x0	Auto CMD12 Not Executed This error occurs when the host controller cannot issue an Auto Cmd12 to stop multiple block data transfer due to some errors. 0 = Disable 1 = Enable

Table 1357: Capabilities 1 and 0 Register
Offset: 0x000D8040

Bit	Field	Type/InitVal	Description
31:30	CFG_SLOT_TYPE	RW 0x0	Slot Type Indicates what type of slot the host controller is connected to. 0 = Removable card slot 1 = Embedded slot for 1 device 2 = Shared bus slot 3 = Reserved
29	ASYNC_INT_SUPPORT	RO 0x1	Asynchronous Interrupt Support 0 = Not supported 1 = Supported
28	SYS_BUS_64_SUPPORT	RO 0x0	64-bit System Bus Support Indicates whether the host controller is capable of 64-bit system bus. 0 = Not supported 1 = Supported
27	Reserved	RSVD 0x0	Reserved
26	VLG_18_SUPPORT	RO 0x1	Voltage Support 1.8V Indicates whether the host controller is capable of 1.8V. 0 = 1.8V not supported. 1 = 1.8V supported.

Copyright © 2015 Marvell

Table 1357: Capabilities 1 and 0 Register (Continued)
Offset: 0x000D8040

Bit	Field	Type/InitVal	Description
25	VLG_30_SUPPORT	RO 0x0	Voltage Support 3.0V Indicates whether the host controller is capable of 3.0V. 0 = 3.0V not supported. 1 = 3.0V supported.
24	VLG_33_SUPPORT	RO 0x1	Voltage Support 3.3V Indicates whether the host controller is capable of 3.3V. 0 = 3.3V not supported. 1 = 3.3V supported.
23	SUS_RES_SUPPORT	RO 0x1	Suspend Resume Support Indicates whether the host controller is capable of suspend/resume commands. 0 = Not supported 1 = Supported
22	SDMA_SUPPORT	RO 0x1	SDMA Support Indicates whether the host controller is capable of SDMA. 0 = Not supported 1 = Supported
21	HI_SPEED_SUPPORT	RO 0x1	High Speed Support Indicates whether the host controller is capable of high speed mode (25-50 MHz). 0 = Not supported 1 = Supported
20	ADMA1_SUPPORT	RO 0x1	ADMA1 Support Indicates whether the host controller is capable of ADMA1. 0 = Not supported 1 = Supported
19	ADMA2_SUPPORT	RO 0x1	ADMA2 Support Indicates whether the host controller is capable of ADMA2. 0 = Not supported 1 = Supported
18	EX_DATA_WIDTH_SUPPORT	RW 0x1	8-bit support. Indicates whether the host controller is capable of 8-bit bus operation. For SDIO the default is 0x0. For eMMC the default is 0x1. 0 = 8-bit not supported. 1 = 8-bit supported
17:16	MAX_BLK_LEN	RO 0x2	Maximum Block Length Maximum block length in bytes. 0x0 = 512 bytes
15:8	BASE_FREQ	RO 0xC8	Base Frequency The base clock frequency for SDCLK. 0xC8 = 200 MHz
7	TIMEOUT_UNIT	RO 0x1	Timeout Unit The base clock unit used to detect timeouts. 0 = kHz 1 = MHz

Copyright © 2015 Marvell

Table 1357: Capabilities 1 and 0 Register (Continued)
Offset: 0x000D8040

Bit	Field	Type/InitVal	Description
6	Reserved	RSVD 0x0	Reserved
5:0	TIMEOUT_FREQ	RO 0x32	Timeout Frequency Indicates the base clock frequency used to detect timeouts. 0x32 = 50 MHz

Table 1358: Capabilities 2 and 3 Register
Offset: 0x000D8044

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28	VLT2_18_SUPPORT	RO 0x1	VLT2_18 Support
27	ADMA3_SUPPORT	RO 0x0	ADMA3 Support
26:24	Reserved	RSVD 0x0	Reserved
23:16	CLOCK_MULTIPLIER	RO 0x0	Clock Multiplier This field indicates clock multiplier value of programmable clock generator. 0x0 = Host controller does not support a programmable clock generator.
15:14	RETUNE_MODES	RO 0x0	Re-Tuning Modes Selects the re-tuning method and limits the maximum data length. 0 = Mode1: Timer 1 = Mode2: Timer and Re-Tuning Request 2 = Mode3: Auto Re-Tuning (for transfer) Timer and Re-Tuning Request. 3 = Reserved
13	SDR50_TUNE	RO 0x1	Use Tuning for SDR50 Mode 0 = No tuning: SDR50 does not require tuning. 1 = Requires tuning: SDR50 requires tuning
12	Reserved	RSVD 0x0	Reserved
11:8	TMR_RETUNE	RO 0xF	Timer Count for Re-Tuning Indicates an initial value of the Re-Tuning Timer for Mode 1 to 3. 0xF = Get information from other source.
7	Reserved	RSVD 0x0	Reserved
6	DRV_TYPE_D	RO 0x1	Driver Type D Support 0 = Not supported 1 = Supported

Table 1358: Capabilities 2 and 3 Register (Continued)
Offset: 0x000D8044

Bit	Field	Type/InitVal	Description
5	DRV_TYPE_C	RO 0x1	Driver Type C Support 0 = Not supported 1 = Supported
4	DRV_TYPE_A	RO 0x1	Driver Type A Support 0 = Not supported 1 = Supported
3	UHSII_SUPPORT	RO 0x1	UHSII Support 0 = Not supported 1 = Supported
2	DDR50_SUPPORT	RO 0x1	DDR50 Support 0 = Not supported 1 = Supported
1	SDR104_SUPPORT	RO 0x1	SDR104 Support 0 = Not supported 1 = Supported
0	SDR50_SUPPORT	RO 0x1	SDR50 Support 0 = Not supported 1 = Supported

Table 1359: Maximum Current 1 and 2 Register
Offset: 0x000D8048

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:16	MAX_CUR_18	RO 0x0	Maximum Current for 1.8V 0x0 = Get information via another method. 0x1 = 4 mA 0x2 = 8 mA ... 0xFF = 1020 mA
15:8	MAX_CUR_30	RO 0x0	Maximum Current for 3.0V 0x0 = Get information via another method. 0x1 = 4 mA 0x2 = 8 mA ... 0xF = 1020 mA
7:0	MAX_CUR_33	RO 0x0	Maximum Current for 3.3V 0x0 = Get information via another method. 0x1 = 4 mA 0x2 = 8 mA ... 0xF = 1020 mA

Copyright © 2015 Marvell

Table 1360: Maximum Current 3 and 4 Register
Offset: 0x000D804C

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	MAX_CUR_18_V2	RO 0x0	Maximum Current for 1.8V 0x0 = Get information via another method. 0x1 = 4 mA 0x2 = 8 mA ... 0xFF = 1020 mA

Table 1361: Force Event Auto Command 12 Error Register
Offset: 0x000D8050

Bit	Field	Type/InitVal	Description
31	F_CRC_STATUS_ERR	0x0	Force Event for CRC Status Error When 1 is written in this field, it sets the <CRC Status Error> field in the Error Interrupt Status Register.
30	F_CPL_TIMEOUT_ERR	0x0	Force Event for CPL Timeout Error When 1 is written in this field, it sets the <Command Completion Signal Timeout Error> field in the Error Interrupt Status Register.
29	F_AXI_RESP_ERR	0x0	Force Event for AXI Response Bit Error When 1 is written in this field, it sets the <AXI Bus Response Error> field in the Error Interrupt Status Register.
28	F_SPI_ERR	0x0	Force Event for SPI Error When 1 is written in this field, it sets the <SPI Mode Error> field in the Error Interrupt Status Register.
27:26	Reserved	RSVD 0x0	Reserved
25	F_ADMA_ERR	0x0	Force Event for ADMA Error When 1 is written in this field, it sets the <ADMA Error> field in the Error Interrupt Status Register.
24	F_ACMD12_ERR	0x0	Force Event for Auto Cmd12 Error When 1 is written in this field, it sets the <Auto CMD12 Error> field in the Error Interrupt Status Register.
23	F_CURRENT_ERR	0x0	Force Event for Current Limit Error When 1 is written in this field, it sets the <Current Limit Error> field in the Error Interrupt Status Register.
22	F_DAT_END_BIT_ERR	0x0	Force Event for Data End Bit Error When 1 is written in this field, it sets the <Read Data End Bit Error> field in the Error Interrupt Status Register.
21	F_DAT_CRC_ERR	0x0	Force Event for Data CRC Error When 1 is written in this field, it sets the <Read Data CRC Error> field in the Error Interrupt Status Register.

Copyright © 2015 Marvell

Table 1361: Force Event Auto Command 12 Error Register (Continued)
Offset: 0x000D8050

Bit	Field	Type/InitVal	Description
20	F_DAT_TO_ERR	0x0	Force Event for Data Timeout Error When 1 is written in this field, it sets the <Read Data CRC Error> field in the Error Interrupt Status Register.
19	F_CMD_INDEX_ERR	0x0	Force Event for Command Index Error When 1 is written in this field, it sets the <Data Timeout Error> field in the Error Interrupt Status Register.
18	F_CMD_END_BIT_ERR	0x0	Force Event for Command End Bit Error When 1 is written in this field, it sets the <Command Index Error> field in the Error Interrupt Status Register.
17	F_CMD_CRC_ERR	0x0	Force Event for Command CRC Error When 1 is written in this field, it sets the <Command CRC Error> field in the Error Interrupt Status Register.
16	F_CMD_TO_ERR	0x0	Force Event for Command Timeout Error When 1 is written in this field, it sets the <Command Timeout Error> field in the Error Interrupt Status Register.
15:8	Reserved	RSVD 0x0	Reserved
7	F_ACMD12_ISSUE_ERR	0x0	Force Event for Command Not Issued by Auto Cmd 12 Error When 1 is written in this field, it sets the <Command Not Issued Due to auto_cmd12 Error> field in the Auto CMD12 Error Status Register.
6:5	Reserved	RSVD 0x0	Reserved
4	F_ACMD_INDEX_ERR	0x0	Force Event for Auto Cmd Index Error When 1 is written in this field, it sets the <Auto CMD Error> field in the Auto CMD Error Status Register.
3	F__ACMD_EBIT_ERR	0x0	Force Event for Auto Cmd End Bit Error When 1 is written in this field, it sets the <Auto CMD End Bit Error> field in the Auto CMD Error Status Register.
2	F_ACMD_CRC_ERR	0x0	Force Event for Auto Cmd CRC Error When 1 is written in this field, it sets the <Auto CMD CRC Error> field in the Auto CMD Error Status Register.
1	F_ACMD_TO_ERR	0x0	Force Event for Auto Cmd Timeout Error When 1 is written in this field, it sets the <Auto CMD Timeout Error> field in the Auto CMD Error Status Register.
0	F_ACMD12_NEXE_ERR	0x0	Force Event for Auto CMD 12 Not Executed Error When 1 is written in this field, it sets the <Auto CMD12 Not Executed> field in the Auto CMD12 Error Status Register.

Table 1362: ADMA Error Status Register
Offset: 0x000D8054

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1362: ADMA Error Status Register (Continued)
Offset: 0x000D8054

Bit	Field	Type/InitVal	Description
2	ADMA_LEN_ERR	RW 0x0	ADMA Length Mismatch Error This error occurs in the following cases: - While Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. - Total data length cannot be divided by the block length.
1:0	ADMA_STATE	RW 0x0	ADMA Error State Indicates the state of ADMA when an error occurred during ADMA transfer. This field never indicates 0x2 because ADMA never stops in that state.

Table 1363: ADMA System Address Register High
Offset: 0x000D8058

Bit	Field	Type/InitVal	Description
31:0	ADMA_SYS_ADDR_H	RW 0x0	ADMA System Address High This register holds byte address of executing command of the Descriptor table. At the start of ADMA, this register should be programmed to point to the start address of the Descriptor table. This register is incremented whenever fetching a descriptor line. When an ADMA Error Interrupt is generated, this register will hold a valid Descriptor address, depending on the ADMA state.

Table 1364: ADMA System Address Register Low
Offset: 0x000D805C

Bit	Field	Type/InitVal	Description
31:0	ADMA_SYS_ADDR_L	RW 0x0	ADMA System Address Low This register holds byte address of executing command of the Descriptor table. At the start of ADMA, this register should be programmed to point to the start address of the Descriptor table. This register is incremented whenever fetching a descriptor line. When an ADMA Error Interrupt is generated, this register will hold a valid Descriptor address, depending on the ADMA state.

Table 1365: Slot Interrupt Status Register
Offset: 0x000D80FC

Bit	Field	Type/InitVal	Description
31:24	VENDOR_VER	RO 0x0	Marvell Version Number
23:16	SD_VER	RO 0x4	SD Host Specification Number 0 = Supports version 1.0 1 = Supports version 2.0 2 = Supports version 3.0: All other values are reserved.
15:4	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1365: Slot Interrupt Status Register (Continued)
Offset: 0x000D80FC

Bit	Field	Type/InitVal	Description
3	SLOT_INT3	RO 0x0	Interrupt Line for Slot 3
2	SLOT_INT2	RO 0x0	Interrupt Line for Slot 2
1	SLOT_INT1	RO 0x0	Interrupt Line for Slot 1
0	SLOT_INT0	RO 0x0	Interrupt Line for Slot 0

Table 1366: SD Extra Parameters Register
Offset: 0x000D8100

Bit	Field	Type/InitVal	Description
31:24	GEN_PAD_CLK_CNT	RW 0x74	Generate Pad Clock Count This field should be used in conjunction with the <Gen_Pad_Clk_On> field. This field configures the number of clock cycles generated on the IO pad. The default value of 0x74 generates 75 clock cycles.
23:7	Reserved	RSVD 0x0	Reserved
6	GEN_PAD_CLK_ON	RW 0x0	Generate Pad Clock This bit should be set in conjunction with the field <Gen_Pad_Clk_Cnt>. Setting this bit to 0x1 generates the programmed number of clock cycles on the IO pad.
5	SQU_FULL_CHK	RW 0x0	SQU Full Check This bit should be set to 0x1 only if using certain segment of memory in SQU in FIFO mode and a Read transaction is going to be performed on the SD device.
4	SQU_EMPTY_CHK	RW 0x0	SQU Empty Check This bit should be set to 0x1 only if using certain segment of memory in SQU in FIFO mode and a Write transaction is going to be performed on the SD device.
3	BOOT_ACK	RW 0x0	Boot ACK If the boot ACK mode is enabled in the MMC device then this field should be written to 0x1 before issuing the alternate boot cmd0.
2:0	Reserved	RSVD 0x0	Reserved

Table 1367: FIFO Parameters Register
Offset: 0x000D8104

Bit	Field	Type/InitVal	Description
31:20	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1367: FIFO Parameters Register (Continued)
Offset: 0x000D8104

Bit	Field	Type/InitVal	Description
19:16	PRE_GATE_CLK_CNT	RW 0x3	Clock Count Before Gate This field controls the amount of clock cycles that are provided before clock gating is enabled on the sd_clk I/O pad.
15	Reserved	RSVD 0x0	Reserved
14	PDLVMC	RW 0x0	PDLVMC This field controls some power down function for the internal 64x64 memory.
13	PDFVSSM	RW 0x0	PDFVSSM This field controls some power down function for the internal 64x64 memory.
12	FORCE_CLK_ON	RW 0x0	Force SD I/O Pad Clock On Setting this field to 0x1 overrides the SD Clock I/O pad clock gate and forces the clock on. This field is meant to be used in conjunction with the <OVRRD_CLK_OEN> field. Bit [11] Bit [12] Setting ---------- ---------- ---------- 0 -- <Force SD I/O Pad Clock On> setting is ignored (not used). 1 0 Clocks are forced off. 1 1 Clocks are forced on.
11	OVRRD_CLK_OEN	RW 0x0	Override Pad Clock Output Enable Setting this field to 0x1 overrides the SD Pad clock output enable. This field is meant to be used in conjunction with the <FORCE_CLK_ON> field. Bit [11] Bit [12] Setting ---------- ---------- ---------- 0 -- <Force SD I/O Pad Clock On> setting is ignored (not used). 1 0 Clocks are forced off. 1 1 Clocks are forced on.
10	Reserved	RSVD 0x0	Reserved
9	CLK_GATE_ON	RW 0x1	Clock Gate On This field affects all clock gates in the SD design if <CLK_GATE_CTL> is set to 1. 0 = Disable clock gate. 1 = Enable clock gate.
8	CLK_GATE_CTL	RW 0x0	Clock Gate Control A register setting of <CLK_GATE_CTL> = 0x1 and <CLK_GATE_ON> = 0x0 is invalid and not allowed. Such a write command is ignored. The <CLK_GATE_ON> and <CLK_GATE_CTL> fields have been provided for software to override the internal clock gating of the SD module. 0 = Disable: Disables software clock gating override. 1 = Enable: Enables software clock gating override.
7	USE_DAT3	RW 0x0	Use DAT[3] This field allows the card detect functionality to be detected, using the DAT[3] pin. 0 = Use dedicated pin. 1 = Use DAT[3].

Table 1367: FIFO Parameters Register (Continued)
Offset: 0x000D8104

Bit	Field	Type/InitVal	Description
6	PDWN	RW 0x0	Power Down Controls the Power Down port on the internal 2 port 64x67 FIFO.
5	FIFO_CS	RW 0x0	FIFO CS Enable This field should be written to 0x1 before toggling the <PDWN> bit.
4	FIFO_CLK	RW 0x0	FIFO Clock Enable This field should be set to 0x1 before toggling the <PDWN> bit.
3:2	WTC	RW 0x2	WTC This field is used for FIFO speed setting.
1:0	RTC	RW 0x1	RTC This field is used for FIFO speed setting.

Table 1368: SPI Mode and Clock and Burst Size Setup Register
Offset: 0x000D8108

Bit	Field	Type/InitVal	Description
31	WR_OSTDG	RW 0x0	Write Outstanding This field controls the outstanding write requests on AXI. 0 = Allow: Allows outstanding write requests on AXI. 1 = Do not allow: Does not allow outstanding write requests on AXI.
30	RD_OSTDG	RW 0x0	Read Outstanding This field controls the outstanding read requests on AXI. 0 = Allow: Allows outstanding read requests on AXI. 1 = Do not allow: Does not allow outstanding read requests on AXI.
29:24	Reserved	RSVD 0x0	Reserved
23	WR_ENDIAN	RW 0x1	Write Endian Defines the format of the data written to the card. 0 = Big Endian 1 = Little Endian
22	RD_ENDIAN	RW 0x0	Read Endian Defines the format of the data read from the card. 0 = Big Endian 1 = Little Endian
21	AXI_NON_POST_WR	RW 0x0	AXI Non-post Write 0 = Last only: Only the last request is issued as non-post write. 1 = All: All AXI master write requests are non-post write.
20	PRIORITY	RW 0x1	AXI Priority Arbitration Request This is a static bit that is passed along as the Most Significant Bit (MSb) of the ID when AXI requests are made. This bit should be changed only before and after completion of a data command and not in between. 0 = Low priority 1 = High priority

Copyright © 2015 Marvell

Table 1368: SPI Mode and Clock and Burst Size Setup Register (Continued)
Offset: 0x000D8108

Bit	Field	Type/InitVal	Description
19:18	DMA_SIZE	RW 0x3	FIFO Threshold This field is used for the FIFO threshold for the internal FSM to generate a DMA request to the AXI master. 0 = 64 bytes 1 = 128 bytes 2 = 192 bytes 3 = 256 bytes
17:16	BRST_SIZE	RW 0x0	DMA Burst Size on the AXI Fabric 0 = 32 bytes burst 1 = 64 bytes burst 2 = 128 bytes burst 3 = Reserved: This values must not be used.
15:13	Reserved	RSVD 0x0	Reserved
12:8	SPI_ERR_TOKEN	RW 0x0	SPI Error Token Received in command response when SPI mode is enabled.
7:1	Reserved	RSVD 0x0	Reserved
0	SPI_EN	RW 0x0	Enable SPI Mode Indicates that SPI mode has been enabled. This causes the host controller to drive the signals on the interface in accordance with the SPI protocol. 0 = Disable 1 = Enable

Table 1369: CE-ATA Register
Offset: 0x000D810C

Bit	Field	Type/InitVal	Description
31	CHK_CPL	RW 0x0	Check Command Completion Signal When this field is set to 1 and the <CEATA_CARD> field is set to 1, an indication is sent to the host controller to check for the command completion signal from the CE-ATA card.
30	SND_CPL	RW 0x0	Send Command Completion Disable Signal When this field is set to 1 and the <CEATA_CARD> field is set to 1, an indication is sent to the host controller to send the command completion disable signal to the CE-ATA card.
29	CEATA_CARD	RW 0x0	CE-ATA Card Mode Enable 0 = Disable: Non CE-ATA card mode 1 = Enable: CE-ATA Card mode
28	MMC_CARD	RW 0x1	MMC Card Enable 0 = Disable: SD Card mode 1 = Enable: MMC Card mode
27	MMC_RESETN	RW 0x0	MMC Resetn This bit controls the value of the pin MMC_RESETN going to the eMMC device.

Table 1369: CE-ATA Register (Continued)
Offset: 0x000D810C

Bit	Field	Type/InitVal	Description
26:23	Reserved	RSVD 0x0	Reserved
22	CPL_COMPLETE	RW 0x0	CPL Complete This bit is set to 1 when a command completion signal is detected and the <CPL_COMPLETE_EN> field was set to 1. This field is cleared by writing 0x1. A write of 0x0 has no effect.
21	CPL_COMPLETE_EN	RW 0x0	CPL Complete Enable This bit, when set to 0x1, enables the CPL_COMPLETE bit.
20	CPL_COMPLETE_INT_EN	RW 0x0	CPL Complete Interrupt Enable An interrupt is generated whenever the <CPL_COMPLETE> field is set.
19	Reserved	RSVD 0x0	Reserved
18	MISC_INT	RW 0x0	Miscellaneous Interrupt This status bit is set to 1 when the programmed number of clocks in SD_CFG FIFO Parameter register <Gen_Pad_Clk_Cnt> field is completed and the <Misc_Int_En> field has been set to 0x1. This field is cleared by writing 0x1. A write of 0x0 has no effect.
17	MISC_INT_EN	RW 0x0	Miscellaneous Interrupt Status Enable Setting this field to 1 enables the Misc_Int bit to be set to 0x1 when the programmed number of clocks have been generated on the pad.
16	MISC_INT_INT_EN	RW 0x0	Miscellaneous Interrupt Interrupt Enable When set to 0x1, an interrupt is generated whenever the <Misc_Int> field is set to 0x1.
15:14	Reserved	RSVD 0x0	Reserved
13:0	CPL_TIMEOUT	RW 0x3fff	Command Completion Signal Timeout Value

Table 1370: Drive Strength Register
Offset: 0x000D8110

Bit	Field	Type/InitVal	Description
31:24	Reserved	RSVD 0x0	Reserved
23:16	ECO_REG	RW 0x0	ECO Spare Register
15:3	Reserved	RSVD 0x0	Reserved
2	CLR_WAKEUP	RW 0x1	Clear Asynchronous Wake-up Event A write of 0x1 to this bit clears the wake-up event. This bit is auto cleared. A write of 0x0 has no effect.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1370: Drive Strength Register (Continued)
Offset: 0x000D8110

Bit	Field	Type/InitVal	Description
1	INAND_SEL	RW 0x0	Inand Select
0	ASYNC_IO_EN	RW 0x0	Async Read Interface Enable This bit enables the asynchronous latching of input data. 0 = Disable: Async interface is disabled. 1 = Enable: Async interface is enabled. The clock used to latch the input data and the internal logic clock are asynchronous.

Table 1371: RX Configuration Register
Offset: 0x000D8114

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25:17	TUNING_DLY_INC	RW 0x0	Tuning Delay Increment When tuning is performed, the host controller takes over the control of the delay value being programmed in the delay element. This field defines the value by which the delay value is incremented for each step.
16:8	SDCLK_DELAY	RW 0x0	SDCLK Delay This field controls the delay value to the delay element.
7:4	Reserved	RSVD 0x0	Reserved
3:2	SDCLK_SEL1	RW 0x0	SDCLK Select 1 This field controls the second mux selects. 0 = Select clock from pad. 1 = Inverted clock: Select inverted clock from ddll. 2 = Select internal clock.: Same as 0x3 3 = Select internal_clock.: Same as 0x2
1:0	SDCLK_SEL0	RW 0x0	SDCLK Select 0 This field controls the first mux selects. 0 = Select clock from pad. 1 = Inverted clock from pad: Select inverted clock from pad 2 = Select internal clock. 3 = Inverted internal clock: Select inverted internal clock.

Table 1372: TX Configuration Register
Offset: 0x000D8118

Bit	Field	Type/InitVal	Description
31	TX_MUX_SEL	RW 0x0	Tx Mux Select This field selects the final clock that drives the data out on the SD bus. When cleared to 0, this field selects the input 200 MHz clock to the module. The data is launched on the bus at the negedge of this clock. This setting is appropriate for all modes.

Copyright © 2015 Marvell

Table 1372: TX Configuration Register (Continued)
Offset: 0x000D8118

Bit	Field	Type/InitVal	Description
30:25	Reserved	RSVD 0x0	Reserved
24:16	TX_HOLD_DELAY1	RW 0x82	Tx Hold Delay 1 This field controls the delay value to the Tx delay element for sdr104 mode.
15:9	Reserved	RSVD 0x0	Reserved
8:0	TX_HOLD_DELAY0	RW 0x112	Tx Hold Delay 0 This field controls the delay value to the Tx delay element for all modes other than the sdr104 mode.

Table 1373: SD Host Controller Version 0 Register
Offset: 0x000D81F8

Bit	Field	Type/InitVal	Description
31:24	SD_VER	RO 0x64	SD Version
23:16	SD_TAPEOUT_SPECIFIC	RO 0x0	Tapeout Specific Value 0
15:8	MMC_VER	RO 0x69	MMC Version
7:0	MMC_TAPEOUT_SPECIFIC	RO 0x0	Tapeout Specific Value

Table 1374: SD Host Controller Version 1 Register
Offset: 0x000D81FC

Bit	Field	Type/InitVal	Description
31:24	SDIO_VER	RO 0x64	SDIO Version
23:16	SDIO_TAPEOUT_SPECIFIC	RO 0x0	Tapeout Specific Value
15:0	CHIP_ID	RO 0x160	Chip ID

A.17.2 Mbus to AXI Bridge Control

Table 1375: Mbus Port Control Register
Offset: 0x000DC000

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1375: Mbus Port Control Register (Continued)
Offset: 0x000DC000

Bit	Field	Type/InitVal	Description
5:1	Mbus Retry Timer	RW 0x1F	Mbus Retry Timer. Start counts upon request and if <Mbus Retry En> is enabled, and the timer expires, a retry respond is sent back over the Mbus.
0	Mbus Retry En	RW 0x0	Applies the Mbus retry response to the read or write request when the <Mbus Retry Timer> is expired. 0 = Retry Disable 1 = Retry Enable

Table 1376: AXI Port Control Register
Offset: 0x000DC004

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8:5	AXI Wr Cache Attr	RW 0x0	AXI Write Cache attribute to be sent from the Bridge towards the unit's AXI Slave port. The cache attribute should be determined according to the AXI 1.0 specification, chapter 5.1, Table 5-1.
4:1	AXI Rd Cache Attr	RW 0x0	AXI Read Cache attribute to be sent from Bridge towards unit's AXI Slave port. The cache attribute should be determined according the AXI 1.0 specification, chapter 5.1, Table 5-1.
0	AXI Wr Strb Err	RW 0x0	When Enabled, sets empty transfer over the AXI WSTRB lines (all equal to 0) when write a erroneous beat was sent from the Mbus side. 0 = Disable: Forwards Erroneous Mbus beats towards the AXI port as is. 1 = Enable: Forwards Erroneous Mbus beats towards the AXI with all WSRTB bits at 0.

A.17.3 AXI to Mbus Bridge Window and Control

Table 1377: Bridge Window Control Register (N=0–7)
Offset: Windows0: 0x000DC080, Windows1: 0x000DC088, Windows2: 0x000DC090, Windows3: 0x000DC098, Windows4: 0x000DC0A0, Windows5: 0x000DC0A8, Windows6: 0x000DC0B0, Windows7: 0x000DC0B8
Offset Formula: 0x000DC080+0x8*N: where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
31:16	Size	RW 0x0	Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 64 KB granularity (for example, a value of 0x00FF specifies 256*64KB = 16 MB). NOTES: A value of 0x0 specifies 64-KB size. Changing the value of this field must be done when <Win En>=0.

Table 1377: Bridge Window Control Register (N=0–7) (Continued)

Offset: Windows0: 0x000DC080, Windows1: 0x000DC088, Windows2: 0x000DC090, Windows3: 0x000DC098, Windows4: 0x000DC0A0, Windows5: 0x000DC0A8, Windows6: 0x000DC0B0, Windows7: 0x000DC0B8

Offset Formula: 0x000DC080+0x8*N: where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
15:8	Attr	RW 0x0	Specifies the target interface attributes associated with this window. See the "Target Interface Configurations" section. Changing the value of this field must be done when <Win En>=0.
7:4	Target	RW 0x0	Specifies the target interface associated with this window. See the "Target Interface Configurations" section. Changing the value of this field must be done when <Win En>=0.
3:1	Reserved	RSVD 0x0	Reserved
0	Win En	RW 0x0	Window0 Enable 0 = Disable: Window is disabled. 1 = Enable: Window is enabled.

Table 1378: Bridge Window Base Register (N=0–7)

Offset: Windows0: 0x000DC084, Windows1: 0x000DC08C, Windows2: 0x000DC094, Windows3: 0x000DC09C, Windows4: 0x000DC0A4, Windows5: 0x000DC0AC, Windows6: 0x000DC0B4, Windows7: 0x000DC0BC

Offset Formula: 0x000DC084+0x8*N: where N (0-7) represents Windows

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register!			
31:16	Base	RW 0x0	Base Address Used with the <Size> field in the Window0 Control Register to set the address window size and location. Corresponds to transaction address[31:16]. Changing the value of this field must be done when <Win0 En>=0
15:0	Reserved	RSVD 0x0	Reserved

Table 1379: Mbus Control Register

Offset: 0x000DC0C0

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8:1	Mbus Read Regret Timer	RW 0x40	Mbus Read Regret Timer. Mbus read request may regret when timer expires and Mbus Regret is Enabled.

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1379: Mbus Control Register (Continued)
Offset: 0x000DC0C0

Bit	Field	Type/InitVal	Description
0	Mbus Read Regret Enable	RW 0x0	Allows the Mbus read request to regret when the Regret Timer expires.

Table 1380: Sync Barrier WindowN Control Register (N=0–1)
Offset: SyncbWin0: 0x000DC0C4, SyncbWin1: 0x000DC0CC
Offset Formula: 0x000DC0C4+0x8*N: where N (0-1) represents SyncbWin

Bit	Field	Type/InitVal	Description
31:2	Sync Barrier WindowN Size	RW 0x0	Sync Barrier Window Size Used with the Base register to set the address window size and location. Must be programmed from LSB to MSB as a sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 4 B granularity (for example, a value of 0x00FF specifies 256*4B = 1KB). NOTES: A value of 0x0 specifies 64-KB size. Changing the value of this field must be done when <Win En>=0
1	Reserved	RSVD 0x0	Reserved
0	Sync Barrier WindowN Enable	RW 0x0	Window Sync Barrier Enable When enabled, data associated with read requests that hit the window will be returned to the requesting CPU after all writes issued by the target unit reach the memory. 0 = Disable: Non blocking 1 = Enable: Blocking

Table 1381: Sync Barrier WindowN Base Register (N=0–1)
Offset: SyncbWin0: 0x000DC0C8, SyncbWin1: 0x000DC0D0
Offset Formula: 0x000DC0C8+0x8*N: where N (0-1) represents SyncbWin

Bit	Field	Type/InitVal	Description
NOTE: If the remap function for this register is not used, the <Remap> field in the Window0 Remap Low Register must be set to same value as the <Base> field in this register!			
31:2	Sync Barrier WindowN Base	RW 0x0	Base Address Used with the <Size> field in the Sync Barrier Window Control Register to set the address window size and location. Corresponds to transaction address[31:2]. Changing the value of this field must be done when <Sync Barrier Enable>=0
1:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.18 Inter-Integrated Circuit (I2C) Registers

The following table provides a summarized list of all registers that belong to the Inter-Integrated Circuit (I2C), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1382: Summary Map Table for the Inter-Integrated Circuit (I2C) Registers

Register Name	Offset	Table and Page
I2C Port		
I2C Slave Address Register (t=0–3)	I2C Port0: 0x00011000, I2C Port1: 0x00011100, I2C Port2: 0x00011200, I2C Port3: 0x00011300	Table 1383, p. 1121
I2C Data Register (t=0–3)	I2C Port0: 0x00011004, I2C Port1: 0x00011104, I2C Port2: 0x00011204, I2C Port3: 0x00011304	Table 1384, p. 1122
I2C Control Register (t=0–3)	I2C Port0: 0x00011008, I2C Port1: 0x00011108, I2C Port2: 0x00011208, I2C Port3: 0x00011308	Table 1385, p. 1122
I2C Baud Rate Register (t=0–3)	I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C	Table 1386, p. 1123
I2C Status Register (t=0–3)	I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C	Table 1387, p. 1123
I2C Soft Reset Register (t=0–3)	I2C Port0: 0x0001101C, I2C Port1: 0x0001111C, I2C Port2: 0x0001121C, I2C Port3: 0x0001131C	Table 1388, p. 1125
I2C Configuration Debug Register	0x0001108C	Table 1389, p. 1125
I2C Initialization Last Data Register	0x00011098	Table 1390, p. 1125
I2C Crossbar Port w Remap Register (w=0–15)	port index0: 0x00013200, port index1: 0x00013204...port index15: 0x0001323C	Table 1391, p. 1125
I2C Bridge		
I2C Bridge Transmit Data Low Register (t=0–3)	I2C Port0: 0x000110C0, I2C Port1: 0x000111C0, I2C Port2: 0x000112C0, I2C Port3: 0x000113C0	Table 1392, p. 1126
I2C Bridge Transmit Data High Register (t=0–3)	I2C Port0: 0x000110C4, I2C Port1: 0x000111C4, I2C Port2: 0x000112C4, I2C Port3: 0x000113C4	Table 1393, p. 1126
I2C Bridge Receive Data Low Register (t=0–3)	I2C Port0: 0x000110C8, I2C Port1: 0x000111C8, I2C Port2: 0x000112C8, I2C Port3: 0x000113C8	Table 1394, p. 1126

Copyright © 2015 Marvell

Table 1382: Summary Map Table for the Inter-Integrated Circuit (I2C) Registers (Continued)

Register Name	Offset	Table and Page
I2C Bridge Receive Data High Register (t=0–3)	I2C Port0: 0x000110CC, I2C Port1: 0x000111CC, I2C Port2: 0x000112CC, I2C Port3: 0x000113CC	Table 1395, p. 1126
I2C Bridge Control Register (t=0–3)	I2C Port0: 0x000110D0, I2C Port1: 0x000111D0, I2C Port2: 0x000112D0, I2C Port3: 0x000113D0	Table 1396, p. 1126
I2C Bridge Status Register (t=0–3)	I2C Port0: 0x000110D4, I2C Port1: 0x000111D4, I2C Port2: 0x000112D4, I2C Port3: 0x000113D4	Table 1397, p. 1127
I2C Bridge Interrupt Cause Register (t=0–3)	I2C Port0: 0x000110D8, I2C Port1: 0x000111D8, I2C Port2: 0x000112D8, I2C Port3: 0x000113D8	Table 1398, p. 1128
I2C Bridge Interrupt Mask Register (t=0–3)	I2C Port0: 0x000110DC, I2C Port1: 0x000111DC, I2C Port2: 0x000112DC, I2C Port3: 0x000113DC	Table 1399, p. 1128
I2C Bridge Timing Gaps Register (t=0–3)	I2C Port0: 0x000110E0, I2C Port1: 0x000111E0, I2C Port2: 0x000112E0, I2C Port3: 0x000113E0	Table 1400, p. 1129

A.18.1 I2C Port

Table 1383: I2C Slave Address Register (t=0–3)
Offset: I2C Port0: 0x00011000, I2C Port1: 0x00011100, I2C Port2: 0x00011200, I2C Port3: 0x00011300
Offset Formula: 0x00011000+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:1	SAddr	RW 0x64	Slave address For a 7-bit slave address, bits [7:1] are the slave address. For a 10-bit address, SAddr[7:3] must be set to 11110 and SAddr[2:1] stands for the two MSB (bits [9:8]) of the 10-bit address.
0	GCE	RW 0x0	General Call Enable If set to 1, the I2C slave interface responds to general call accesses.

Table 1384: I2C Data Register (t=0–3)

Offset: I2C Port0: 0x00011004, I2C Port1: 0x00011104, I2C Port2: 0x00011204, I2C Port3: 0x00011304
Offset Formula: 0x00011004+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:8	Reserved	RO 0x0	Reserved
7:0	Data	RW 0x0	Data/Address byte to be transmitted by the I2C master or slave, or data byte received In the case of the Address byte, bit [0] is the Read/Write Command bit.

Table 1385: I2C Control Register (t=0–3)

Offset: I2C Port0: 0x00011008, I2C Port1: 0x00011108, I2C Port2: 0x00011208, I2C Port3: 0x00011308
Offset Formula: 0x00011008+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:8	Reserved	RO 0x0	Reserved
7	IntEn	RW 0x0	Interrupt Enable When set to 1, an interrupt is generated each time the interrupt flag is set.
6	I2CEn	RW 0x0	I2C Enable 0 = Ignore: I2C_SDA and I2C_SCK inputs are ignored. The I2C slave does not respond to any address on the bus. 1 = Respond: The I2C slave responds to calls to its slave address, and to general calls if enabled.
5	Start	RW 0x0	Start When set to 1, the I2C master initiates a start condition on the bus, when the bus is free, or a repeated start condition, if the master already drives the bus. The bit is set only. It is cleared by I2C hardware after a start condition is driven on the bus.
4	Stop	RW 0x0	Stop When set to 1, the I2C master initiates a stop condition on the bus. The bit is set only. It is cleared by I2C hardware after a stop condition is driven on the bus.
3	IFlg	RW 0x0	Interrupt Flag If any of the status codes other than 0xF8 are set, the I2C hardware sets the bit to 1. If set to 1 and I2C interrupts are enabled through bit [7], an interrupt is asserted. Cleared by a CPU core write of 0.
2	ACK	RW 0x0	Acknowledge When set to 1, the I2C drives an acknowledge bit on the bus in response to a received address (slave mode), or in response to a data received (read data in master mode, write data in slave mode). For a master to signal a I2C target with a read of last data, the CPU core must clear this bit (generating no acknowledge bit on the bus). For the slave to respond, this bit must always be set back to 1.
1:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1386: I2C Baud Rate Register (t=0–3)
Offset: I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C
Offset Formula: 0x0001100C+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:7	Reserved	RSVD 0x0	Reserved
6:3	M	WO 0x4	See exact frequency calculation in the I2C section. Write only.
2:0	N	WO 0x5	See exact frequency calculation in the I2C section. Write only.

Table 1387: I2C Status Register (t=0–3)
Offset: I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C
Offset Formula: 0x0001100C+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1387: I2C Status Register (t=0–3) (Continued)

Offset: I2C Port0: 0x0001100C, I2C Port1: 0x0001110C, I2C Port2: 0x0001120C, I2C Port3: 0x0001130C

Offset Formula: 0x0001100C+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
7:0	Stat	RO 0x0	I2C Status 0 = BusError 8 = StartTransmit: Start condition transmitted. 16 = StartTransmitRepeat: Repeated start condition transmitted. 24 = AddWrAck: Address plus write bit transmitted, acknowledge received. 32 = AddWrNoAck: Address plus write bit transmitted, acknowledge not received. 40 = MstrTxAck: Master transmitted data byte, acknowledge received. 48 = MstrTxNoAck: Master transmitted data byte, acknowledge not received. 56 = MstrLostArb: Master lost arbitration during address or data transfer. 64 = AddRdAck: Address plus read bit transmitted, acknowledge received. 72 = AddRdNoAck: Address +read bit transmitted, acknowledge not received. 80 = MstrRxAck: Master received read data, acknowledge transmitted. 88 = MstrRxNoAck: Master received read data, acknowledge not transmitted. 96 = SlRcdAddAck: Slave received slave address, acknowledge transmitted. 104 = MstrArbLost: Master lost arbitration during address transmit, address is targeted to the slave (write access), acknowledge transmitted. 112 = GnrRxAck: General call received, acknowledge transmitted. 120 = MstrArbLost: Master lost arbitration during address transmit, general call address received, acknowledge transmitted. 128 = SlRxWrDataAck: Slave received write data after receiving slave address, acknowledge transmitted. 136 = SlRxWrDataSlaveNoAck: Slave received write data after receiving slave address, acknowledge not transmitted. 144 = SlRxWrDataAck: Slave received write data after receiving general call, acknowledge transmitted. 152 = SlRxWrDataNoAck: Slave received write data after receiving general call, acknowledge not transmitted. 160 = SLRxStopStart: Slave received stop or repeated start condition. 168 = SlRxAddAck: Slave received address plus read bit, acknowledge transmitted. 176 = MstrArbLost: Master lost arbitration during address transmit, address is targeted to the slave (read access), acknowledge transmitted. 184 = SlTxRdDataAck: Slave transmitted read data, acknowledge received. 192 = SlTxRdDataNoAck: Slave transmitted read data, acknowledge not received. 200 = SlTxRdByteAck: Slave transmitted last read byte, acknowledge received. 208 = SecondAddWrBitTxAck: Second address plus write bit transmitted, acknowledge received. 216 = SecondAddWrBitTxNoAck: Second address plus write bit transmitted, acknowledge not received. 224 = SecondAddRdBitTxAck: Second address plus read bit transmitted, acknowledge received. 232 = SecondAddRdBitTxNoAck: Second address + read bit transmitted, acknowledge not received. 248 = NotRelevant: No relevant status. Interrupt flag is kept 0.

Copyright © 2015 Marvell

Table 1388: I2C Soft Reset Register (t=0–3)
Offset: I2C Port0: 0x0001101C, I2C Port1: 0x0001111C, I2C Port2: 0x0001121C, I2C Port3: 0x0001131C
Offset Formula: 0x0001101C+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:0	Rst	WO 0x0	Write Only Write to this register resets the I2C logic and sets all I2C registers to their reset values.

Table 1389: I2C Configuration Debug Register
Offset: 0x0001108C

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x33	Reserved
15:8	Attr	RW 0x0	Defines the target unit attributes for the I2C serial initialization and debug port access to the external interface (non-internal registers).
7:4	Reserved	RSVD 0x1	Reserved
3:0	UnitID	RW 0x0	Defines the unit ID for the I2C serial initialization and debug port access to external interface (non-internal registers).

Table 1390: I2C Initialization Last Data Register
Offset: 0x00011098

Bit	Field	Type/InitVal	Description
31:0	Last	RW 0xFFFFFFFF	Serial initialization last data indication. Can be changed during the serial initialization sequence to terminate the initialization with a different pattern.

Table 1391: I2C Crossbar Port w Remap Register (w=0–15)
Offset: port index0: 0x00013200, port index1: 0x00013204...port index15: 0x0001323C
Offset Formula: 0x00013200+0x4*w: where w (0-15) represents port index

Bit	Field	Type/InitVal	Description
31:17	Reserved	RSVD 0x0	Reserved
16	Internal	RW 0x0	When remap is enabled, and AD[31:28] = w, this value replaces the Internal bit
15:13	Reserved	RSVD 0x0	Reserved
12:4	Target_Specific	RW 0x0	When remap is enabled, and AD[31:28] = w, this value replaces the target specific bits (CBE[20:12])
3:0	Address_MSBs	RW 0x0	When remap is enabled, and AD[31:28] = w, this value replaces the upper 4 bits of the address when accessing the xBar

Copyright © 2015 Marvell

A.18.2 I2C Bridge

Table 1392: I2C Bridge Transmit Data Low Register (t=0–3)
Offset: I2C Port0: 0x000110C0, I2C Port1: 0x000111C0, I2C Port2: 0x000112C0, I2C Port3: 0x000113C0
Offset Formula: 0x000110C0+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:0	TxDataLow	RW 0x0	4 Low data bytes to be transmitted by the I2C master.

Table 1393: I2C Bridge Transmit Data High Register (t=0–3)
Offset: I2C Port0: 0x000110C4, I2C Port1: 0x000111C4, I2C Port2: 0x000112C4, I2C Port3: 0x000113C4
Offset Formula: 0x000110C4+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:0	TxDataHigh	RW 0x0	4 High data bytes to be transmitted by the I2C master.

Table 1394: I2C Bridge Receive Data Low Register (t=0–3)
Offset: I2C Port0: 0x000110C8, I2C Port1: 0x000111C8, I2C Port2: 0x000112C8, I2C Port3: 0x000113C8
Offset Formula: 0x000110C8+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:0	RxDataLow	RO 0x0	4 Low data bytes received from I2C slave.

Table 1395: I2C Bridge Receive Data High Register (t=0–3)
Offset: I2C Port0: 0x000110CC, I2C Port1: 0x000111CC, I2C Port2: 0x000112CC, I2C Port3: 0x000113CC
Offset Formula: 0x000110CC+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:0	RxDataHigh	RO 0x0	4 High data bytes received from I2C slave.

Table 1396: I2C Bridge Control Register (t=0–3)
Offset: I2C Port0: 0x000110D0, I2C Port1: 0x000111D0, I2C Port2: 0x000112D0, I2C Port3: 0x000113D0
Offset Formula: 0x000110D0+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:21	Reserved	RSVD 0x0	Reserved
20	RepeatedStart	RW 0x0	If set, and <Wr> and <Rd> are set, perform a read transaction after the write without a stop bit between them.
19	BridgeEn	RW 0x0	When set and the <Rd>/<Wr> bits are set, starts an I2C bridge transaction.

Copyright © 2015 Marvell

Table 1396: I2C Bridge Control Register (t=0–3) (Continued)
Offset: I2C Port0: 0x000110D0, I2C Port1: 0x000111D0, I2C Port2: 0x000112D0, I2C Port3: 0x000113D0
Offset Formula: 0x000110D0+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
18:16	RxSize	RW 0x0	This field indicates the number of bytes that needs to be received. 0 = 1 byte 1 = 2 bytes 2 = 3 bytes 3 = 4 bytes 4 = 5 bytes 5 = 6 bytes 6 = 7 bytes 7 = 8 bytes
15:13	TxSize	RW 0x0	This field indicates the number of bytes that needs to be transmitted. 0 = 1 byte 1 = 2 bytes 2 = 3 bytes 3 = 4 bytes 4 = 5 bytes 5 = 6 bytes 6 = 7 bytes 7 = 8 bytes
12	ExtendAddr	RW 0x0	Extended Slave Address Set to 0x1, if the Slave Address is 10 bits. 0 = 7-bit Address 1 = 10-bit Address
11:2	Addr	RW 0x0	I2C Slave Address
1	Rd	RW 0x0	Read Transaction When set to 0x1, perform a read transaction. If the <Wr> bit is set, also perform a read after the write.
0	Wr	RW 0x0	Write Transaction When set to 0x1, perform a write transaction. If the <Rd> bit is set, also perform a read after the write.

Table 1397: I2C Bridge Status Register (t=0–3)
Offset: I2C Port0: 0x000110D4, I2C Port1: 0x000111D4, I2C Port2: 0x000112D4, I2C Port3: 0x000113D4
Offset Formula: 0x000110D4+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:14	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1397: I2C Bridge Status Register (t=0–3) (Continued)
Offset: I2C Port0: 0x000110D4, I2C Port1: 0x000111D4, I2C Port2: 0x000112D4, I2C Port3: 0x000113D4
Offset Formula: 0x000110D4+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
13:11	ErrorState	RO 0x0	Indicates the reason for the error. **NOTE:** Other values are reserved. 0 = No Error 1 = Start Error: Status after <START> bit set indicates an error. 2 = Addr Error: Status after address phase indicates an error. 3 = Extended Addr Error: Status after extended address phase indicates an error. 4 = Tx Error: Status after transmit data indicates an error. 5 = Rx Error: Status after received data indicates an error.
10:8	BytesLeft	RO 0x0	Indicate the number of bytes left to transmit/receive.
7:1	Reserved	RSVD 0x0	Reserved
0	Error	RO 0x0	Set to 1 when an error is detected during the transaction. 0 = No error 1 = Error

Table 1398: I2C Bridge Interrupt Cause Register (t=0–3)
Offset: I2C Port0: 0x000110D8, I2C Port1: 0x000111D8, I2C Port2: 0x000112D8, I2C Port3: 0x000113D8
Offset Formula: 0x000110D8+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	Interrupt	RW 0x0	The interrupt sets when a transaction ends or an error was detected.

Table 1399: I2C Bridge Interrupt Mask Register (t=0–3)
Offset: I2C Port0: 0x000110DC, I2C Port1: 0x000111DC, I2C Port2: 0x000112DC, I2C Port3: 0x000113DC
Offset Formula: 0x000110DC+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	InterruptMask	RW 0x0	When set the interrupt is masked.

Copyright © 2015 Marvell

Table 1400: I2C Bridge Timing Gaps Register (t=0–3)

Offset: I2C Port0: 0x000110E0, I2C Port1: 0x000111E0, I2C Port2: 0x000112E0, I2C Port3: 0x000113E0

Offset Formula: 0x000110E0+t*0x100 : where t (0-3) represents I2C Port

Bit	Field	Type/InitVal	Description
31:25	Reserved	RSVD 0x0	Reserved
24:20	StopToStartGap	RW 0x0	A delay of 2^Value in terms of core_clk between the stop bit and the start bit.
19:15	AckToStopGap	RW 0x0	A delay of 2^Value in terms of core_clk between the ACK received and the stop bit.
14:10	AckToNextByteGap	RW 0x0	A delay of 2^Value in terms of core_clk between the ACK received and the next byte transmitted/received.
9:5	StartToAddrGap	RW 0x0	A delay of 2^Value in terms of core_clk between the start bit and the address phase.
4:0	AckToStartGap	RW 0x0	A delay of 2^Value in terms of core_clk between the ACK received and the start bit.

Copyright © 2015 Marvell

A.19 Universal Asynchronous Receiver/Transmitter (UART) Registers

The following table provides a summarized list of all registers that belong to the Universal Asynchronous Receiver/Transmitter (UART), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1401: Summary Map Table for the Universal Asynchronous Receiver/Transmitter (UART) Registers

Register Name	Offset	Table and Page
Universal Asynchronous Receive/Transmit (UART)		
Divisor Latch Low (DLL) Register (n=0–3)	UART0: 0x00012000, UART1: 0x00012100,	Table 1402, p. 1131
Transmit Holding (THR) Register (n=0–3)	UART0: 0x00012000, UART1: 0x00012100,	Table 1403, p. 1131
Receive Buffer (RBR) Register (n=0–3)	UART0: 0x00012000, UART1: 0x00012100,	Table 1404, p. 1131
Interrupt Enable (IER) Register (n=0–3)	UART0: 0x00012004, UART1: 0x00012104,	Table 1405, p. 1132
Divisor Latch High (DLH) Register (n=0–3)	UART0: 0x00012004, UART1: 0x00012104,	Table 1406, p. 1133
FIFO Control (FCR) Register (n=0–3)	UART0: 0x00012008, UART1: 0x00012108,	Table 1407, p. 1133
Interrupt Identity (IIR) Register (n=0–3)	UART0: 0x00012008, UART1: 0x00012108,	Table 1408, p. 1134
Line Control (LCR) Register (n=0–3)	UART0: 0x0001200C, UART1: 0x0001210C,	Table 1409, p. 1134
Modem Control (MCR) Register (n=0–3)	UART0: 0x00012010, UART1: 0x00012110,	Table 1410, p. 1135
Line Status (LSR) Register (n=0–3)	UART0: 0x00012014, UART1: 0x00012114,	Table 1411, p. 1136
Modem Status (MSR) Register (n=0–3)	UART0: 0x00012018, UART1: 0x00012118,	Table 1412, p. 1137
Scratch Pad (SCR) Register (n=0–3)	UART0: 0x0001201C, UART1: 0x0001211C,	Table 1413, p. 1137
FIFO Access (FAR) Register (n=0–3)	UART0: 0x00012070, UART1: 0x00012170,	Table 1414, p. 1138
UART Status (USR) Register (n=0–3)	UART0: 0x0001207C, UART1: 0x0001217C,	Table 1415, p. 1138
Halt TX (HTX) Register (n=0–3)	UART0: 0x000120A4, UART1: 0x000121A4,	Table 1416, p. 1139

Copyright © 2015 Marvell

A.19.1 Universal Asynchronous Receive/Transmit (UART)

Table 1402: Divisor Latch Low (DLL) Register (n=0–3)
Offset: UART0: 0x00012000, UART1: 0x00012100,
Offset Formula: 0x00012000+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	DivLatchLow	WO 0x0	Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DivLatchRdWrt bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that when the Divisor Latch Registers (DLL and DLH) are set to zero, the baud clock is disabled and no serial communications occur.

Table 1403: Transmit Holding (THR) Register (n=0–3)
Offset: UART0: 0x00012000, UART1: 0x00012100,
Offset Formula: 0x00012000+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	TxHold	WO 0x0	The THR is a write-only register that contains data to be transmitted from the serial port. Any time that the Transmit Holding Register Empty <THRE> bit of the Line Status Register (LSR) is set, data can be written to the LSR <TxEmpty> to be transmitted from the serial port. If FIFOs are not enabled and THRE is set, writing a single word to the THR resets the THRE and any additional writes to the THR before the <THRE> is set again causes the THR data to be overwritten. If FIFOs are enabled and <THRE> is set, up to 16 words of data may be written to the THR before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost.

Table 1404: Receive Buffer (RBR) Register (n=0–3)
Offset: UART0: 0x00012000, UART1: 0x00012100,
Offset Formula: 0x00012000+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1404: Receive Buffer (RBR) Register (n=0–3) (Continued)
Offset: UART0: 0x00012000, UART1: 0x00012100,
Offset Formula: 0x00012000+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
7:0	RxBuf	RO 0x0	The RBR is a read-only register that contains the data byte transmitted to the serial port.The data in this register is valid only if the LSR <DataRxStat> bit in the Line Status Register (LSR) is set. In the non-FIFO mode (fifo_mode = 0), the data in the RBR must be read before the next data arrives; otherwise it will be overwritten, resulting in an overrun error. In the FIFO mode (fifo_mode = 1), 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 word arrives, then the data already in the FIFO will be preserved but any incoming data will be lost.

Table 1405: Interrupt Enable (IER) Register (n=0–3)
Offset: UART0: 0x00012004, UART1: 0x00012104,
Offset Formula: 0x00012004+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7	PTIME	RW 0x0	Programmable THRE Interrupt Mode Enable that can be written to only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. 0 = Disable 1 = Enable
6:4	Reserved	RSVD 0x0	Reserved
3	ModStatIntEn	RW 0x0	Enable Modem Status Interrupt (EDSSI) This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt 0 = Disable: Disable interrupt 1 = Enable: Enable interrupt
2	RxLineStatIntEn	RW 0x0	Enable Receiver Line Status Interrupt (ELSI) This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = Disable: Disable interrupt 1 = Enable: Enable interrupt
1	TxHoldIntEn	RW 0x0	Enable Transmitter Holding Register Empty Interrupt (ETBEI) This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = Disable: Disable interrupt 1 = Enable: Enable interrupt
0	RxDataIntEn	RW 0x0	Enable Received Data Available Interrupt (ERBFI). This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = Disable: Disable interrupt 1 = Enable: Enable interrupt

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 1406: Divisor Latch High (DLH) Register (n=0–3)
Offset: UART0: 0x00012004, UART1: 0x00012104,
Offset Formula: 0x00012004+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	DivLatchHigh	WO 0x0	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.This register may only be accessed when the DivLatchRdWrt bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the core_clk frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (core clock freq) / (16 * divisor). Note that when the Divisor Latch Registers (DLL and DLH) are set to zero, the baud clock is disabled and no serial communications occur.

Table 1407: FIFO Control (FCR) Register (n=0–3)
Offset: UART0: 0x00012008, UART1: 0x00012108,
Offset Formula: 0x00012008+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:6	RxTrigger	WO 0x0	Receive Trigger This is used to select the trigger level in the receiver FIFOat which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the RTSn signal is de-asserted. 0 = 1Byte: 1 byte in Rx FIFO 1 = 4Byte: 4 bytes in Rx FIFO 2 = 8Byte: 8 bytes in Rx FIFO 3 = 14Byte: 14 bytes Rx FIFO
5:4	Tx Empty Trigger	RW 0x0	TX Empty Trigger (TET). This is used to select the empty threshold level at which the THRE Interrupts are generated. 0 = Empty: All Tx FiFO is not occupied 1 = 2Bytes: 14 bytes in Tx FiFO are not occupied 2 = 4Byes: 12 bytes in Tx FiFO are not occupied 3 = 8Bytes: 8 bytes in Tx FiFO are not occupied
3	Reserved	RSVD 0x0	Reserved
2	TxFIFOReset	WO 0x0	Transmit FIFO reset This resets the control portion of the Tx FIFO and treats the FIFO as empty. 0 = NoFlushTx: Do not flush data from the transmit FIFO 1 = FlushTx: Flush data from the transmit FIFO
1	RxFIFOReset	WO 0x0	Receive FIFO reset This resets the control portion of the Rx FIFO and treats the FIFO as empty 0 = NoFlushRx: Do not flush data from the receive FIFO. 1 = FlushRx: Flush data from the receive FIFO.

Copyright © 2015 Marvell

Table 1407: FIFO Control (FCR) Register (n=0–3) (Continued)
Offset: UART0: 0x00012008, UART1: 0x00012108,
Offset Formula: 0x00012008+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
0	FIFOEn	WO 0x0	Enable transmit and receive FIFOs. This register controls the read and write data FIFO operation. 0 = Disable: Disable FIFOs 1 = Enable: Enable FIFOs

Table 1408: Interrupt Identity (IIR) Register (n=0–3)
Offset: UART0: 0x00012008, UART1: 0x00012108,
Offset Formula: 0x00012008+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RO 0x0	Reserved
7:6	FIFOEn	RO 0x0	FIFO Enable 0 = Disable: FIFOs are disabled (default in FIFO mode) 3 = Enable: FIFOs are enabled
5:4	Reserved	RSVD 0x0	Reserved
3:0	InterruptID	RO 0x0	Interrupt ID (IID) This indicates the highest priority pending interrupt. The interrupt priorities are split into four levels. Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt. 0 = Modem: Modem Status Changed 1 = NoInterruptPen: No interrupt pending 2 = THREmpty 4 = RxData: Received Data available 6 = RxStatus: Receiver Status 7 = BusyDetect 12 = TO: Character Time Out

Table 1409: Line Control (LCR) Register (n=0–3)
Offset: UART0: 0x0001200C, UART1: 0x0001210C,
Offset Formula: 0x0001200C+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7	DivLatchRdWrt	RW 0x0	This bit must be set to address (reading and writing) of the Divisor Latch Low (DLL) Register and Divisor Latch High (DLH) Register) to set the baud rate of the UART. This bit must be cleared to address the Receive Buffer Register (RBR), Transmit Holding Register (THR) and the Interrupt Enable Register (IER). The bit is writeable only when UART is not busy (USR[0] is zero)

Copyright © 2015 Marvell

Table 1409: Line Control (LCR) Register (n=0–3) (Continued)
Offset: UART0: 0x0001200C, UART1: 0x0001210C,
Offset Formula: 0x0001200C+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
6	Break	RW 0x0	The <Break> bit sends a break signal by holding the SOUT line low until the <Break> bit is reset. 0 = No Signal: Do not send a break signal 1 = Signal: Send a break signal
5	Reserved	RSVD 0x0	Reserved
4	EPS	RW 0x0	Even or Odd Parity Select The bit is writeable only when UART is not busy (USR[0] is zero) 0 = Odd: Odd parity 1 = Even: Even parity
3	PEN	RW 0x0	Parity Enable The bit is writeable only when UART is not busy (USR[0] is zero) 0 = Disable 1 = Enable
2	Stop	RW 0x0	Stop Bits Transmitted 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. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is 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. The bit is writeable only when UART is not busy (USR[0] is zero) 0 = 1Bits 1 = 2Bits
1:0	WLS	RW 0x0	This is used to select the number of data bits per character that the peripheral transmits and receives The bit is writeable only when UART is not busy (USR[0] is zero) 0 = 5Bits 1 = 6Bits 2 = 7Bits 3 = 8Bits

Table 1410: Modem Control (MCR) Register (n=0–3)
Offset: UART0: 0x00012010, UART1: 0x00012110,
Offset Formula: 0x00012010+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:6	Reserved	RW 0x0	Reserved
5	AFCE	RW 0x0	Auto Flow Control Enable. 0 = False: Auto Flow Control Mode disabled 1 = True: Auto Flow Control Mode enabled

Copyright © 2015 Marvell

Table 1410: Modem Control (MCR) Register (n=0–3) (Continued)
Offset: UART0: 0x00012010, UART1: 0x00012110,
Offset Formula: 0x00012010+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
4	Loopback	RW 0x0	Loopback The <Loopback> bit loops the data on the sout line back to the sin line. In this mode all the interrupts are fully functional. This feature is used for diagnostic purposes. 0 = NoLoopback 1 = Loopback
3:2	Reserved	RSVD 0x0	Reserved
1	RTS	RW 0x0	Request To Send The <RTS> bit is inverted and then drives the corresponding UA_RTSn output.
0	Reserved	RSVD 0x0	Reserved

Table 1411: Line Status (LSR) Register (n=0–3)
Offset: UART0: 0x00012014, UART1: 0x00012114,
Offset Formula: 0x00012014+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7	RxFIFOErr	ROC 0x0	This bit is only active when FIFOs are enabled. It is set when there is at least one parity error, framing error, or break indication in the FIFO. This bit is cleared when the Line Control Register (LCR) is read.
6	TxEmpty	RO 0x0	Transmitter Empty bit In FIFO mode, this bit is set whenever the Transmit Holding Register (THR), the Transmitter Shift Register, and the FIFO are all empty.
5	THRE	RO 0x0	Transmit Holding If the <THRE> bit is set, the device can accept a new character for transmission. If interrupts are enabled, it can cause an interrupt to occur when data from the Transmit Holding Register (THR) is transmitted to the transmit shift register. 0 = Closed: Do not accept a new character for transmission 1 = Open: Accept a new character for transmission
4	BI	ROC 0x0	The <BI> bit is set whenever the serial input (sin) is held in a logic 0 state for longer than the sum of start time + data bits + parity + stop bits. In the FIFO mode, the BI indication is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the Line Control Register (LCR) clears the <BI>.
3	FrameErr	ROC 0x0	Frame Error The FE bit flags a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error comes to the head of the FIFO. The OE, PE and FE bits are reset when a read on the Line Control Register (LCR) is performed.

Copyright © 2015 Marvell

Table 1411: Line Status (LSR) Register (n=0–3) (Continued)
Offset: UART0: 0x00012014, UART1: 0x00012114,
Offset Formula: 0x00012014+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
2	ParErr	ROC 0x0	Parity Error This bit indicates a parity error in the receiver if the <PEN> bit in the Line Control Register (LCR) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the bad parity comes to the head of the FIFO.
1	OverRunErr	ROC 0x0	Overrun Error 0 = NoError: No overrun 1 = Error: Overrun error has occurred
0	DataRxStat	RO 0x0	Receive buffer status This bit is cleared when the Receive Buffer Register (RBR) is read. 0 = Empty: No characters in the receive buffer or FIFO. 1 = NotEmpty: Receive buffer or FIFO contains at least one character. A read operation of the Receiver Buffer Register clears this bit.

Table 1412: Modem Status (MSR) Register (n=0–3)
Offset: UART0: 0x00012018, UART1: 0x00012118,
Offset Formula: 0x00012018+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved
4	CTS	RO 0x0	The CTS Modem Status bit--<CTS> --contains information on the current state of the modem control line. <CTS> is the compliment of UA_CTSn. In Loopback Mode, <CTS> is the same as Modem Control Register (MCR) bit[1] <RTS>.
3:1	Reserved	RSVD 0x0	Reserved
0	DCTS	RO 0x0	The <DCTS> bit records whether the modem control line UA_CTSn has changed since the last time the CPU core read the MSR. In Loopback mode, <DCTS> reflects changes on Modem Control Register (MCR) bit[1] <RTS>.

Table 1413: Scratch Pad (SCR) Register (n=0–3)
Offset: UART0: 0x0001201C, UART1: 0x0001211C,
Offset Formula: 0x0001201C+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:0	Scratch	RW 0x0	The SCR register is an 8-bit read/write register for programmers to use as a temporary storage space.

Copyright © 2015 Marvell

Table 1414: FIFO Access (FAR) Register (n=0–3)
Offset: UART0: 0x00012070, UART1: 0x00012170,
Offset Formula: 0x00012070+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	FAR	RW 0x0	FiFO Access Register. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

Table 1415: UART Status (USR) Register (n=0–3)
Offset: UART0: 0x0001207C, UART1: 0x0001217C,
Offset Formula: 0x0001207C+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved
4	RFF	RO 0x0	Receive FIFO Full. This bit is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	RFNE	RO 0x0	Receive FIFO Not Empty. This bit is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	TFE	RO 0x1	Transmit FIFO Empty. This bit is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	TFNF	RO 0x1	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.

Copyright © 2015 Marvell

Table 1415: UART Status (USR) Register (n=0–3) (Continued)
Offset: UART0: 0x0001207C, UART1: 0x0001217C,
Offset Formula: 0x0001207C+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
0	BUSY	RO 0x0	UART Busy. This bit indicates that a serial transfer is in progress. When cleared, indicates that the UART unit is idle or inactive. **NOTE:** It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the UART unit has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the unit. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. 0 = Idle: Idle or Inactive 1 = Busy: Actively transferring data

Table 1416: Halt TX (HTX) Register (n=0–3)
Offset: UART0: 0x000120A4, UART1: 0x000121A4,
Offset Formula: 0x000120A4+n*0x100 : where n (0-3) represents UART

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	Halt TX	RW 0x0	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are enabled. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are not enabled, the setting of the halt TX register has no effect on operation.

Copyright © 2015 Marvell

A.20 General Purpose Input/Output (GPIO) Registers

The following table provides a summarized list of all registers that belong to the General Purpose Input/Output (GPIO), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1417: Summary Map Table for the General Purpose Input/Output (GPIO) Registers

Register Name	Offset	Table and Page
GPIO Configuration and Control		
GPIO_0_31_DataOut Register	0x00018100	Table 1418, p. 1141
GPIO_0_31_Data Out Enable Control Register	0x00018104	Table 1419, p. 1141
GPIO_0_31_Blink Enable Register	0x00018108	Table 1420, p. 1141
GPIO_0_31_Data In Polarity Register	0x0001810C	Table 1421, p. 1141
GPIO_0_31_Data In Register	0x00018110	Table 1422, p. 1142
GPIO_0_31_Interrupt Cause Register	0x00018114	Table 1423, p. 1142
GPIO_0_31_Interrupt Mask Register	0x00018118	Table 1424, p. 1142
GPIO_0_31_Interrupt Level Mask Register	0x0001811C	Table 1425, p. 1142
GPIO_0_31_Blink_Counter_Select Register	0x00018120	Table 1426, p. 1143
GPIO_0_31_Control_Set Register	0x00018128	Table 1427, p. 1143
GPIO_0_31_Control_Clear Register	0x0001812C	Table 1428, p. 1143
GPIO_0_31_ Data_Out_Set Register	0x00018130	Table 1429, p. 1143
GPIO_0_31_Data_Out_Clear Register	0x00018134	Table 1430, p. 1143
GPIO_32_59_DataOut Register	0x00018140	Table 1431, p. 1144
GPIO_32_59_Data Out Enable Control Register	0x00018144	Table 1432, p. 1144
GPIO_32_59_Blink Enable Register	0x00018148	Table 1433, p. 1144
GPIO_32_59_Data In Polarity Register	0x0001814C	Table 1434, p. 1144
GPIO_32_59_Data In Register	0x00018150	Table 1435, p. 1144
GPIO_32_59_Interrupt Cause Register	0x00018154	Table 1436, p. 1145
GPIO_32_59_Interrupt Mask Register	0x00018158	Table 1437, p. 1145
GPIO_32_59_Interrupt Level Mask Register	0x0001815C	Table 1438, p. 1145
GPIO_32_59_Blink_Counter_Select Register	0x00018160	Table 1439, p. 1146
GPIO_32_59_Control_Set Register	0x00018168	Table 1440, p. 1146
GPIO_32_59_Control_Clear Register	0x0001816C	Table 1441, p. 1146
GPIO_32_59_Data_Out_Set Register	0x00018170	Table 1442, p. 1146
GPIO_32_59 Data Out Clear Register	0x00018174	Table 1443, p. 1147
GPIO Blink Counter A is ON Duration Register	0x000181C0	Table 1444, p. 1147
GPIO Blink Counter A is OFF Duration Register	0x000181C4	Table 1445, p. 1147
GPIO Blink Counter B is ON Duration Register	0x000181C8	Table 1446, p. 1147
GPIO Blink Counter B is OFF Duration Register	0x000181CC	Table 1447, p. 1147
GPIO Interrupt		
CPU<m>_GPIO_0_31_Interrupt Cause Register (m=0–1)	CPU_Num0: 0x00018800, CPU_Num1: 0x00018804	Table 1448, p. 1148
CPU<m>_GPIO_0_31_Interrupt Mask Register (m=0–1)	CPU_Num0: 0x00018810, CPU_Num1: 0x00018814	Table 1449, p. 1148

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1417: Summary Map Table for the General Purpose Input/Output (GPIO) Registers (Continued)

Register Name	Offset	Table and Page
CPU<m>_GPIO_0_31_Interrupt Level Mask Register (m=0–1)	CPU_Num0: 0x00018820, CPU_Num1: 0x00018824	Table 1450, p. 1148
CPU<m>_GPIO_32_59_Interrupt Cause Register (m=0–1)	CPU_Num0: 0x00018840, CPU_Num1: 0x00018844	Table 1451, p. 1148
CPU<m>_GPIO_32_59_Interrupt Mask Register (m=0–1)	CPU_Num0: 0x00018850, CPU_Num1: 0x00018854	Table 1452, p. 1149
CPU<m>_GPIO_32_59_Interrupt Level Mask Register (m=0–1)	CPU_Num0: 0x00018860, CPU_Num1: 0x00018864	Table 1453, p. 1149

A.20.1 GPIO Configuration and Control

Table 1418: GPIO_0_31_DataOut Register
Offset: 0x00018100

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_DataOut	RW 0x0	GPIO Output Pins Value One bit for each GPIO pin.

Table 1419: GPIO_0_31_Data Out Enable Control Register
Offset: 0x00018104

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_DOutEn	RW 0xFFFFFFFF	GPIO Port Output Enable This field is active low. Data is driven when the corresponding bit value is 0.

Table 1420: GPIO_0_31_Blink Enable Register
Offset: 0x00018108

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_DBlink	RW 0x0	GPIO Data Blink When set and the corresponding bit in GPIO Data Out Enable Control register is enabled, the GPIO pin blinks every ~100 ms (a period of 2^24 TCLK clocks).

Table 1421: GPIO_0_31_Data In Polarity Register
Offset: 0x0001810C

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_DataInActLow	RW 0x0	GPIO Data in Active Low When set to 1, the GPIO Data In register reflects the inverted value of the corresponding pin.

Copyright © 2015 Marvell

Table 1422: GPIO_0_31_Data In Register
Offset: 0x00018110

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_DIn	RO 0x0	Each bit in this field reflects the value of the corresponding GPIO pin. If the corresponding bit in the GPIO Data In Polarity register is cleared to 0, the bit reflects the value of the pin with no change. If the corresponding bit in the GPIO Data In Polarity register is set to 1, the bit reflects the pin's inverted value.

Table 1423: GPIO_0_31_Interrupt Cause Register
Offset: 0x00018114

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_Int	RW0C 0x0	A bit in this field is set on transition of the corresponding bit in the <GPIODIn> field in the GPIO Data In register, from 0 to 1.

Table 1424: GPIO_0_31_Interrupt Mask Register
Offset: 0x00018118

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_IntEdgeMask	RW 0x0	GPIO Interrupt Edge Sensitive Mask The mask bit for each cause bit in the <GPIOInt> field of the GPIO Interrupt Cause register. The mask only affects assertion of the interrupt bits in Main Interrupt Cause register. It does not affect the setting of bits in the GPIO Interrupt Cause register. 0 = Interrupt mask; 1 = Interrupt enable;

Table 1425: GPIO_0_31_Interrupt Level Mask Register
Offset: 0x0001811C

Bit	Field	Type/InitVal	Description
To set an edge sensitive interrupt, set the corresponding bit in the GPIO Interrupt Mask Register. To set a level sensitive interrupt, set the corresponding bit in the GPIO Interrupt Level Mask Register.			
31:0	GPIO_0_31_IntLeveIMask	RW 0x0	GPIO Interrupt Level Sensitive Mask The mask bit for each bit in the <GPIODIn> field of the GPIO Data In register. The mask only affects assertion of the Interrupt bit in Main Interrupt Cause register. It does not affect the value of bits in the GPIO Data In register. 0=Interrupt mask; 1=Interrupt enable;

Copyright © 2015 Marvell

Table 1426: GPIO_0_31_Blink_Counter_Select Register
Offset: 0x00018120

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_Blink_Counter_Select	RW 0x0	For each GPIO, this bit selects according to which counter it will blink (if the Blink Enable bit is set to 0x1): 0x0 = Blink Counter A 0x1 = Blink Counter B

Table 1427: GPIO_0_31_Control_Set Register
Offset: 0x00018128

Bit	Field	Type/InitVal	Description
31:0	GPIOD_0_31_OutEnSet	WO 0x0	Write 1 to a bit, sets the corresponding bit in GPIO Data Out Enable register. Write 0 has no affect. Read from this register returns 0.

Table 1428: GPIO_0_31_Control_Clear Register
Offset: 0x0001812C

Bit	Field	Type/InitVal	Description
31:0	GPIODOutEn_0_31_Clear	WO 0x0	Write 1 to a bit, clears the corresponding bit in GPIO Data Out Enable register. Write 0 has no affect. Read from this register returns 0.

Table 1429: GPIO_0_31_ Data_Out_Set Register
Offset: 0x00018130

Bit	Field	Type/InitVal	Description
31:0	GPIO0_31_DOutEnSet	WO 0x0	Write 1 to a bit, sets the corresponding bit in GPIO Data Out register. Write 0 has no affect. Read from this register returns 0.

Table 1430: GPIO_0_31_Data_Out_Clear Register
Offset: 0x00018134

Bit	Field	Type/InitVal	Description
31:0	GPIO_0_31_DOutClear	WO 0x0	Write 1 to a bit, clears the corresponding bit in GPIO Data Out register. Write 0 has no affect. Read from this register returns 0.

Copyright © 2015 Marvell

Table 1431: GPIO_32_59_DataOut Register
Offset: 0x00018140

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_Data Out	RW 0x0	GPIO Output Pins Value One bit for each GPIO pin.

Table 1432: GPIO_32_59_Data Out Enable Control Register
Offset: 0x00018144

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_DOut En	RW 0xFFFFFFF	GPIO Port Output Enable This field is active low. Data is driven when the corresponding bit value is 0.

Table 1433: GPIO_32_59_Blink Enable Register
Offset: 0x00018148

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_DBlin k	RW 0x0	GPIO Data Blink When set and the corresponding bit in GPIO Data Out Enable Control register is enabled, the GPIO pin blinks every ~100 ms (a period of 2^24 TCLK clocks).

Table 1434: GPIO_32_59_Data In Polarity Register
Offset: 0x0001814C

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_DatalnActLow	RW 0x0	GPIO Data in Active Low When set to 1, the GPIO Data In register reflects the inverted value of the corresponding pin.

Table 1435: GPIO_32_59_Data In Register
Offset: 0x00018150

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1435: GPIO_32_59_Data In Register (Continued)
Offset: 0x00018150

Bit	Field	Type/InitVal	Description
27:0	GPIO_32_59_DIn	RO 0x0	Each bit in this field reflects the value of the corresponding GPIO pin. If the corresponding bit in the GPIO Data In Polarity register is cleared to 0, the bit reflects the value of the pin with no change. If the corresponding bit in the GPIO Data In Polarity register is set to 1, the bit reflects the pin's inverted value.

Table 1436: GPIO_32_59_Interrupt Cause Register
Offset: 0x00018154

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_Int	RW0C 0x0	A bit in this field is set on transition of the corresponding bit in the <GPIODIn> field in the GPIO Data In register, from 0 to 1.

Table 1437: GPIO_32_59_Interrupt Mask Register
Offset: 0x00018158

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_IntEdgeMask	RW 0x0	GPIO Interrupt Edge Sensitive Mask The mask bit for each cause bit in the <GPIOInt> field of the GPIO Interrupt Cause register. The mask only affects assertion of the interrupt bits in Main Interrupt Cause register. It does not affect the setting of bits in the GPIO Interrupt Cause register. 0 = Interrupt mask; 1 = Interrupt enable;

Table 1438: GPIO_32_59_Interrupt Level Mask Register
Offset: 0x0001815C

Bit	Field	Type/InitVal	Description
To set an edge sensitive interrupt, set the corresponding bit in the GPIO Interrupt Mask Register. To set a level sensitive interrupt, set the corresponding bit in the GPIO Interrupt Level Mask Register.			
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_IntLevelMask	RW 0x0	GPIO Interrupt Level Sensitive Mask The mask bit for each bit in the <GPIODIn> field of the GPIO Data In register. The mask only affects assertion of the Interrupt bit in Main Interrupt Cause register. It does not affect the value of bits in the GPIO Data In register. 0=Interrupt mask; 1=Interrupt enable;

Copyright © 2015 Marvell

Table 1439: GPIO_32_59_Blink_Counter_Select Register
Offset: 0x00018160

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_Blink_Counter_Select	RW 0x0	For each GPIO, this bit selects according to which counter it will blink (if the Blink Enable bit is set to 0x1): 0x0 = Blink Counter A 0x1 = Blink Counter B

Table 1440: GPIO_32_59_Control_Set Register
Offset: 0x00018168

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIOD_32_59_OutEnSet	WO 0x0	Write 1 to a bit, sets the corresponding bit in GPIO Data Out Enable register. Write 0 has no affect. Read from this register returns 0.

Table 1441: GPIO_32_59_Control_Clear Register
Offset: 0x0001816C

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIODOutEn_32_59_Clear	WO 0x0	Write 1 to a bit, clears the corresponding bit in GPIO Data Out Enable register. Write 0 has no affect. Read from this register returns 0.

Table 1442: GPIO_32_59_Data_Out_Set Register
Offset: 0x00018170

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_DOutSet	WO 0x0	Write 1 to a bit, sets the corresponding bit in GPIO Data Out register. Write 0 has no affect. Read from this register returns 0.

Copyright © 2015 Marvell

Table 1443: GPIO_32_59 Data Out Clear Register
Offset: 0x00018174

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	GPIO_32_59_DOut Clear	WO 0x0	Write 1 to a bit, clears the corresponding bit in GPIO Data Out register. Write 0 has no affect. Read from this register returns 0.

Table 1444: GPIO Blink Counter A is ON Duration Register
Offset: 0x000181C0

Bit	Field	Type/InitVal	Description
31:0	CounterAOnDur	RW 0x1000000	Counter A ON duration in terms of core clock cycles.

Table 1445: GPIO Blink Counter A is OFF Duration Register
Offset: 0x000181C4

Bit	Field	Type/InitVal	Description
31:0	CounterAOffDur	RW 0x1000000	Blink Counter A OFF duration in terms of core clock cycles.

Table 1446: GPIO Blink Counter B is ON Duration Register
Offset: 0x000181C8

Bit	Field	Type/InitVal	Description
31:0	CounterBOnDur	RW 0x1000000	Blink Counter B ON duration in terms of core clock cycles.

Table 1447: GPIO Blink Counter B is OFF Duration Register
Offset: 0x000181CC

Bit	Field	Type/InitVal	Description
31:0	CounterBOffDur	RW 0x1000000	Blink Counter B OFF duration in terms of core clock cycles.

Copyright © 2015 Marvell

A.20.2 GPIO Interrupt

Table 1448: CPU<m>_GPIO_0_31_Interrupt Cause Register (m=0–1)
Offset: CPU_Num0: 0x00018800, CPU_Num1: 0x00018804
Offset Formula: 0x00018800+0x4*m: where m (0-1) represents CPU_Num

Bit	Field	Type/InitVal	Description
31:0	CPU<m>_GPIO_0_31_Int	RW0C 0x0	A bit in this field is set on transition of the corresponding bit in the <GPIODIn> field in the GPIO Data In register, from 0 to 1.

Table 1449: CPU<m>_GPIO_0_31_Interrupt Mask Register (m=0–1)
Offset: CPU_Num0: 0x00018810, CPU_Num1: 0x00018814
Offset Formula: 0x00018810+0x4*m: where m (0-1) represents CPU_Num

Bit	Field	Type/InitVal	Description
31:0	CPU<m>_GPIO_0_31_IntEdgeMask	RW 0x0	GPIO Interrupt Edge Sensitive Mask The mask bit for each cause bit in the <GPIOInt> field of the GPIO Interrupt Cause register. The mask only affects assertion of the interrupt bits in Main Interrupt Cause register. It does not affect the setting of bits in the GPIO Interrupt Cause register. 0 = Interrupt mask; 1 = Interrupt enable;

Table 1450: CPU<m>_GPIO_0_31_Interrupt Level Mask Register (m=0–1)
Offset: CPU_Num0: 0x00018820, CPU_Num1: 0x00018824
Offset Formula: 0x00018820+0x4*m: where m (0-1) represents CPU_Num

Bit	Field	Type/InitVal	Description
To set an edge sensitive interrupt, set the corresponding bit in the GPIO Interrupt Mask Register. To set a level sensitive interrupt, set the corresponding bit in the GPIO Interrupt Level Mask Register.			
31:0	CPU<m>_GPIO_0_31_IntLevelMask	RW 0x0	GPIO Interrupt Level Sensitive Mask The mask bit for each bit in the <GPIODIn> field of the GPIO Data In register. The mask only affects assertion of the Interrupt bit in Main Interrupt Cause register. It does not affect the value of bits in the GPIO Data In register. 0=Interrupt mask; 1=Interrupt enable;

Table 1451: CPU<m>_GPIO_32_59_Interrupt Cause Register (m=0–1)
Offset: CPU_Num0: 0x00018840, CPU_Num1: 0x00018844
Offset Formula: 0x00018840+0x4*m: where m (0-1) represents CPU_Num

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	CPU<m>_GPIO_32_59_Int	RW0C 0x0	A bit in this field is set on transition of the corresponding bit in the <GPIODIn> field in the GPIO Data In register, from 0 to 1.

Copyright © 2015 Marvell

Table 1452: CPU<m>_GPIO_32_59_Interrupt Mask Register (m=0–1)
Offset: CPU_Num0: 0x00018850, CPU_Num1: 0x00018854
Offset Formula: 0x00018850+0x4*m: where m (0-1) represents CPU_Num

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27:0	CPU<m>_GPIO_32_59_IntEdgeMask	RW 0x0	GPIO Interrupt Edge Sensitive Mask The mask bit for each cause bit in the <GPIOInt> field of the GPIO Interrupt Cause register. The mask only affects assertion of the interrupt bits in Main Interrupt Cause register. It does not affect the setting of bits in the GPIO Interrupt Cause register. 0 = Interrupt mask; 1 = Interrupt enable;

Table 1453: CPU<m>_GPIO_32_59_Interrupt Level Mask Register (m=0–1)
Offset: CPU_Num0: 0x00018860, CPU_Num1: 0x00018864
Offset Formula: 0x00018860+0x4*m: where m (0-1) represents CPU_Num

Bit	Field	Type/InitVal	Description
To set an edge sensitive interrupt, set the corresponding bit in the GPIO Interrupt Mask Register. To set a level sensitive interrupt, set the corresponding bit in the GPIO Interrupt Level Mask Register.			
31:28	Reserved	RSVD 0x0	Reserved
27:0	CPU<m>_GPIO_32_59_IntLevelMask	RW 0x0	GPIO Interrupt Level Sensitive Mask The mask bit for each bit in the <GPIODIn> field of the GPIO Data In register. The mask only affects assertion of the Interrupt bit in Main Interrupt Cause register. It does not affect the value of bits in the GPIO Data In register. 0=Interrupt mask; 1=Interrupt enable;

A.21 Real-Time Clock (RTC) Registers

The following table provides a summarized list of all registers that belong to the Real-Time Clock (RTC), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1454: Summary Map Table for the Real-Time Clock (RTC) Registers

Register Name	Offset	Table and Page
RTC Status Register	0x000A3800	Table 1455, p. 1150
RTC Interrupt 1 Configuration Register	0x000A3804	Table 1456, p. 1150
RTC Interrupt 2 Configuration Register	0x000A3808	Table 1457, p. 1151
RTC Time Register	0x000A380C	Table 1458, p. 1151
RTC Alarm 1 Register	0x000A3810	Table 1459, p. 1152
RTC Alarm 2 Register	0x000A3814	Table 1460, p. 1152
RTC Clock Correction Register	0x000A3818	Table 1461, p. 1152
RTC Test Configuration Register	0x000A381C	Table 1462, p. 1152

Table 1455: RTC Status Register
Offset: 0x000A3800

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x0	Reserved
1	Alarm2	RW1C 0x0	Alarm 2 Flag. When alarm 2 is enabled and the alarm event occurs, this bit is set. The bit is cleared by writing 1 to it.
0	Alarm1	RW1C 0x0	Alarm 1 Flag. When alarm 1 is enabled and the alarm event occurs, this bit is set. The bit is cleared by writing 1 to it.

Table 1456: RTC Interrupt 1 Configuration Register
Offset: 0x000A3804

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved
5:2	Int1 Freq	RW 0x0	Interrupt 1 Alarm Frequency. If Int1 Freq En is enabled, sets the frequency of the RTC Alarm 1 Configuration as follows: Int1 Freq[0] sets a 1 Hz frequency. Int1 Freq[1] sets a 2 Hz frequency. Int1 Freq[2] sets a 4 Hz frequency. Int1 Freq[3] sets an 8 Hz frequency. If more than one bit is set, the resulting waveform at the RTC Alarm 1 Configuration register output is the logic OR of the enabled waveform.

Copyright © 2015 Marvell

Table 1456: RTC Interrupt 1 Configuration Register (Continued)
Offset: 0x000A3804

Bit	Field	Type/InitVal	Description
1	Int1 Freq En	RW 0x0	Int1 Alarm Frequency Enable. If set, the Int1 Alarm is toggled as described in Int1 Freq when the programmed alarm time and the time value match.
0	Int1 Alarm En	RW 0x0	Interrupt 1 Alarm Enable. When Int1 Alarm En (alarm enable) is set, the RTC Alarm 1 output is asserted when the programmed alarm time and the time value match.

Table 1457: RTC Interrupt 2 Configuration Register
Offset: 0x000A3808

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved
5:2	Int2 Freq	RW 0x0	Int2 Alarm Frequency. If Int2 Freq En is enabled, sets the frequency of the RTC Alarm 2 as follows: Int2 Freq[0] sets a 1 Hz frequency. Int2 Freq[1] sets a 2 Hz frequency. Int2 Freq[2] sets a 4 Hz frequency. Int2 Freq[3] sets an 8 Hz frequency. If more than one bit is set, the resulting waveform at the RTC_ALARM2 output is the logic OR of the enabled waveform
1	Int2 Freq En	RW 0x0	Int2 Alarm Frequency Enable. If set, the Int2 Alarm is toggled as described in Int2 Freq when the programmed alarm time and the time value match.
0	Int2 Alarm En	RW 0x0	Interrupt 2 Alarm Enable. When Int2 Alarm En (alarm enable) is set, the RTC Alarm 2 output is asserted when the programmed alarm time and the time value match.

Table 1458: RTC Time Register
Offset: 0x000A380C

Bit	Field	Type/InitVal	Description
31:0	Time	RW 0x0	Time Register. This register contains the actual time value of the RTC. Reading this register return the RTC time value. Writing to this register allows programming the RTC time. **NOTE:** The actual write to the timer occurs with the second following the 32-kHz clock edge (rising) after writing Time[7:0]. An internal handshake mechanism synchronizes this write operation and also resets the divider at this time.

Copyright © 2015 Marvell

Table 1459: RTC Alarm 1 Register
Offset: 0x000A3810

Bit	Field	Type/InitVal	Description
31:0	Alarm1	RW 0x0	Alarm 1. When the time matches the programmed alarm value and the alarm is enabled in the corresponding interrupt configuration register (RTC_Interrupt x Configuration), the alarm port goes high for 1 second. The alarm is also signaled by a flag in the RTC Status register.

Table 1460: RTC Alarm 2 Register
Offset: 0x000A3814

Bit	Field	Type/InitVal	Description
31:0	Alarm2	RW 0x0	Alarm 2. When the time matches the programmed alarm value and the alarm is enabled in the corresponding interrupt configuration register (RTC Interrupt x Configuration), the alarm port goes high for 1 second. The alarm is also signaled by a flag in the RTC Status register.

Table 1461: RTC Clock Correction Register
Offset: 0x000A3818

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	Correct Mode	RW 0x0	Correction Mode 0 = Low 1 = High
14:0	Correct Value	RW 0x0	Correction Value

Table 1462: RTC Test Configuration Register
Offset: 0x000A381C

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved
4	Func Test	RW 0x0	Functional Test Acceleration
3:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.22 Power Management Unit Registers

The following table provides a summarized list of all registers that belong to the Power Management Unit, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1463: Summary Map Table for the Power Management Unit Registers

Register Name	Offset	Table and Page
PMU Power Domain Delay Select Register	0x0001C010	Table 1464, p. 1153
PMU Power Up Delay 0 Register	0x0001C014	Table 1465, p. 1153
PMU Power Up Delay 1 Register	0x0001C018	Table 1466, p. 1154
PMU DFS Control 1 Register	0x0001C054	Table 1467, p. 1154
PMU Activate I/F Control Register	0x0001C090	Table 1468, p. 1154

Table 1464: PMU Power Domain Delay Select Register
Offset: 0x0001C010

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x0	Reserved
2	PmuPdDelaySel2	RW 0x0	Controls which delay value will be defined as the delay between the power down control signal de-assertion, and the actual power stabilization. 0 = Delay0: Use the delay value specified by PMU Power Up Delay 0 register. 1 = Delay1: Use the delay value specified by PMU Power Up Delay 1 register.
1	PmuPdDelaySel1	RW 0x0	Controls which delay value will be defined as the delay between the power down control signal de-assertion, and the actual power stabilization. 0 = Delay0: Use the delay value specified by PMU Power Up Delay 0 register. 1 = Delay1: Use the delay value specified by PMU Power Up Delay 1 register.
0	Reserved	RSVD 0x0	Reserved

Table 1465: PMU Power Up Delay 0 Register
Offset: 0x0001C014

Bit	Field	Type/InitVal	Description
31:0	PmuPowerUpDelay0	RW 0x0100	Controls the delay between the power down signal de-assertion and the actual power stabilization. The delay value is counted in core clock cycles. For each power domain, the user can specify whether to use the Delay0 or Delay1 values.

Table 1466: PMU Power Up Delay 1 Register
Offset: 0x0001C018

Bit	Field	Type/InitVal	Description
31:0	PmuPowerUpDelay 1	RW 0xa	Controls the delay between the power down signal de-assertion and the actual power stabilization. The delay value is counted in core clock cycles. For each power domain, the user can specify whether to use the Delay0 or Delay1 values.

Table 1467: PMU DFS Control 1 Register
Offset: 0x0001C054

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:24	DfsInitRatio1	RW SAR	Saves the Initial Reset-Strap divider ratio. (Which is a function of "CPU Subsystem Clock Frequency Options" reset configurations)
23:16	DfsRatio1	RW 0x0	Defines the required ratio the DFS flow will change the divider to. The number specifies the decimal division of the CPU PLL VCO frequency. (The VCO frequency is a function of "CPU Subsystem Clock Frequency Options" reset configurations)
15:0	Reserved	RSVD 0x406	Reserved

Table 1468: PMU Activate I/F Control Register
Offset: 0x0001C090

Bit	Field	Type/InitVal	Description
31:18	Reserved	RSVD 0x0	Reserved
17	PmuDfsOvrdEn	RW 0x0	This field effects the source of the CPU clock divider ratio. It can be driven from the S@R function, at Start-Of-Day, or from the PMU, in case you want to support CPU DFS. 0 = OvrdDis: The CPU divider ratio is driven by the S@R function. 1 = OvrdEn: The CPU divider ratio is driven by the PMU.
16:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.23 Common PHY Registers

The following table provides a summarized list of all registers that belong to the Common PHY, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1469: Summary Map Table for the Common PHY Registers

Register Name	Offset	Table and Page
Common PHY<n> Configuration 1 Register (n=0–5)	PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8	Table 1470, p. 1156
Common PHY<n> Configuration 2 Register (n=0–5)	PHY_NUM0: 0x00018304, PHY_NUM1: 0x0001832C, PHY_NUM2: 0x00018354, PHY_NUM3: 0x0001837C, PHY_NUM4: 0x000183A4, PHY_NUM5: 0x000183CC	Table 1471, p. 1160
Common PHY<n> Configuration 4 Register (n=0–5)	PHY_NUM0: 0x0001830C, PHY_NUM1: 0x00018334, PHY_NUM2: 0x0001835C, PHY_NUM3: 0x00018384, PHY_NUM4: 0x000183AC, PHY_NUM5: 0x000183D4	Table 1472, p. 1161
Common PHY<n> Configuration 5 Register (n=0–5)	PHY_NUM0: 0x00018310, PHY_NUM1: 0x00018338, PHY_NUM2: 0x00018360, PHY_NUM3: 0x00018388, PHY_NUM4: 0x000183B0, PHY_NUM5: 0x000183D8	Table 1473, p. 1162
Common PHY<n> Configuration 6 Register (n=0–5)	PHY_NUM0: 0x00018314, PHY_NUM1: 0x0001833C, PHY_NUM2: 0x00018364, PHY_NUM3: 0x0001838C, PHY_NUM4: 0x000183B4, PHY_NUM5: 0x000183DC	Table 1474, p. 1164
Common PHY<n> Status 1 Register (n=0–5)	PHY_NUM0: 0x00018318, PHY_NUM1: 0x00018340, PHY_NUM2: 0x00018368, PHY_NUM3: 0x00018390, PHY_NUM4: 0x000183B8, PHY_NUM5: 0x000183E0	Table 1475, p. 1164
Common PHY<n> Status 2 Register (n=0–5)	PHY_NUM0: 0x0001831C, PHY_NUM1: 0x00018344, PHY_NUM2: 0x0001836C, PHY_NUM3: 0x00018394, PHY_NUM4: 0x000183BC, PHY_NUM5: 0x000183E4	Table 1476, p. 1166

Copyright © 2015 Marvell

Table 1469: Summary Map Table for the Common PHY Registers (Continued)

Register Name	Offset	Table and Page
Common PHY<n> Status 3 Register (n=0–5)	PHY_NUM0: 0x00018320, PHY_NUM1: 0x00018348, PHY_NUM2: 0x00018370, PHY_NUM3: 0x00018398, PHY_NUM4: 0x000183C0, PHY_NUM5: 0x000183E8	Table 1477, p. 1167
Common PHY<n> Status 4 Register (n=0–5)	PHY_NUM0: 0x00018324, PHY_NUM1: 0x0001834C, PHY_NUM2: 0x00018374, PHY_NUM3: 0x0001839C, PHY_NUM4: 0x000183C4, PHY_NUM5: 0x000183EC	Table 1478, p. 1168
Common PHYs Selectors Register	0x000183FC	Table 1479, p. 1168

Table 1470: Common PHY<n> Configuration 1 Register (n=0–5)
Offset: PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8
Offset Formula: 0x00018300+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31	MacPhyTxElecIdle	RW 0x0	TX Output to Electrical Idle. See the description in the PHY specifications. **NOTE:** When the PHY is working with the PCIe or USB MAC, this signal is driven by the MAC.
30	RX INIT	RW 0x0	Receiver Initialization. This pulse triggers the PHY to start an acquisition for phase and/or DFE/FFE. RX_INIT must not be de-asserted before RX_INIT_DONE is asserted. NOTE: When the Common PHY is connected to a SATA MAC, this configuration is driven by the MAC internal registers. 0 = Valid 1 = Common Mode

Copyright © 2015 Marvell

Table 1470: Common PHY<n> Configuration 1 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8
Offset Formula: 0x00018300+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
29:26	Gen TX	RW 0x0	Receiver Generation Select SATA: 0x0 = 1.5 Gbps 0x1 = 3.0 Gbps 0x2 = 6.0 Gbps PCIe: 0x0 = 2.5 Gbps 0x1 = 5.0 Gbps USB: 0x0 = Reserved 0x1 = 5.0 Gbps QSGMII/ SGMII: 0x6 = 1.25 Gbps 0x8 = 3.125 Gbps **NOTE:** - Do not use settings not listed here. - When the Common PHY is connected to a SATA MAC, this configuration is driven by the MACs internal registers.

Table 1470: Common PHY<n> Configuration 1 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8
Offset Formula: 0x00018300+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
25:22	Gen RX	RW 0x0	Receiver Generation Select SATA: 0x0 = 1.5 Gbps 0x1 = 3.0 Gbps 0x2 = 6.0 Gbps PCIe: 0x0 = 2.5 Gbps 0x1 = 5.0 Gbps USB: 0x0 = Reserved 0x1 = 5.0 Gbps QSGMII / SGMII: 0x6 = 1.25 Gbps 0x8 = 3.125 Gbps **NOTE:** - Do not use settings not listed here. - When the Common PHY is connected to a SATA MAC, this configuration is driven by the MACs internal registers.
21:20	TXDEEMPH	RW 0x0	Transmitter De-Emphasis. NOTES: 1. If the PHY is connected to the PCIe MAC, the LSB of this configuration will be overridden by the PCIe MAC. 2. If the PHY is connected to the USB3 MAC, the configuration will be overridden by the USB3 MAC. 0 = 6 dB de-emphasis 1 = 3_5 dB de-emphasis 2 = No de-emphasis 3 = Reserved
19	Pin TX Idle	RW 0x0	TX Driver Control Signal. NOTE: When the Common PHY is connected to a SATA MAC, this configuration is driven by the MAC internal register file. 0 = Valid: Driver output valid 1 = Common Mode: Driver at common mode voltage varies by the transmitter amplitude settings.

Copyright © 2015 Marvell

Table 1470: Common PHY<n> Configuration 1 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8
Offset Formula: 0x00018300+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
18	Power Up TX	RW 0x1	Power Up Transmitter. NOTE: When the Common PHY is connected to a SATA MAC, this configuration is driven by the MAC internal register file. 0 = Power Down 1 = Power Up
17	Power Up RX	RW 0x1	Power Up Receiver NOTE: When the Common PHY is connected to a SATA MAC, this configuration is driven by the MAC internal register file. 0 = Power Down 1 = Power Up
16	Power Up PLL	RW 0x1	Power Up PLL NOTE: When the Common PHY is connected to a SATA MAC, this configuration is driven by the MAC internal register file. 0 = Power Down 1 = Power Up
15	PHY Mode	RW 0x0	Select PHY Operating Mode **NOTE:** If the Common PHY is used in SATA or SGMII mode, this field is not applicable (NA). 0 = PCI Express Mode 1 = USB Mode
14	PHY Power On Reset	RW 0x0	Power On Reset. **NOTE:** Relevant for PIPE mode only (should be 0x0 in case of PHY mode). Reset signal, active low. This signal resets the entire PIPE, including the register settings. 0 = Reset 1 = Normal
13	PHY Core RSTn	RW 0x0	Core Reset **NOTE:** Relevant for PIPE mode only (should be 0x0 in case of PHY mode). 0 = Reset 1 = Normal
12	PHY Reset Core	RW 0x1	PHY Reset Core **NOTE:** Relevant for PIPE mode only (should be 0x0 in case of PHY mode). 0 = Normal 1 = Reset
11	PHY Software Reset	RW 0x1	PHY Reset Pin This signal resets the entire PHY, including the register settings. 0 = Normal: PHY in Normal mode 1 = Reset: PHY in Reset mode

Copyright © 2015 Marvell

Table 1470: Common PHY<n> Configuration 1 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018300, PHY_NUM1: 0x00018328, PHY_NUM2: 0x00018350, PHY_NUM3: 0x00018378, PHY_NUM4: 0x000183A0, PHY_NUM5: 0x000183C8
Offset Formula: 0x00018300+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
10:3	Reserved	RSVD 0x0	Reserved
2	Pipe Select	RW 0x0	USB 3.0/PCIe PIPE Interface Select. 0 = SATASGMII: SATA or SGMII Interface 1 = USB3 PCIe: USB 3.0/PCIe Interface
1	Power Up IVREF	RW 0x0	Power up current and voltage reference. 0 = Power Down 1 = Power Up
0	RefClkDis	RW 0x0	RefClk Disable 0 = Enable: Enable PHY digital reference clock. 1 = Disable: Request to disable PHY digital reference clock.

Table 1471: Common PHY<n> Configuration 2 Register (n=0–5)
Offset: PHY_NUM0: 0x00018304, PHY_NUM1: 0x0001832C, PHY_NUM2: 0x00018354, PHY_NUM3: 0x0001837C, PHY_NUM4: 0x000183A4, PHY_NUM5: 0x000183CC
Offset Formula: 0x00018304+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x0	Reserved
11	MacPhyTxSwing	RW 0x0	PIPE Interface: Controls transmitter voltage swing level. **NOTE:** When the PHY is operating with the PCIe or USB MAC, this signal is driven by the MAC. 0 = Full swing 1 = Low swing: Low swing
10:8	MacPhyTxMargin	RW 0x0	PIPE Interface: Selects transmitter voltage level. See the PHY specifications for additional information. **NOTE:** When the PHY is operating with the PCIe or USB MAC, this signal is driven by the MAC.
7	MacPhyTxDetectRx LoopBack	RW 0x0	PIPE Interface TX Detect RX Loopback This signal instructs the PHY to begin receiver detection or being loopback. **NOTE:** When the PHY is operating with the PCIe or USB MAC, this signal is driven by the MAC.
6	MacPhyTxCmnMod eDis	RW 0x0	PIPE Interface TX Common Mode Disable 0 = Enable: Enables TX common mode circuit. 1 = Disable: Disables TX common mode circuit.

Copyright © 2015 Marvell

Table 1471: Common PHY<n> Configuration 2 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018304, PHY_NUM1: 0x0001832C, PHY_NUM2: 0x00018354, PHY_NUM3: 0x0001837C, PHY_NUM4: 0x000183A4, PHY_NUM5: 0x000183CC
Offset Formula: 0x00018304+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
5	MacPhyRxPolarity	RW 0x0	PIPE Interface Polarity Inversion on Received Data **NOTE:** When the PHY is operating with the PCIe or USB MAC, this signal is driven by the MAC. 0 = No polarity inversion: PHY performs no polarity inversion. 1 = Polarity inversion: PHY performs polarity inversion.
4	MacPhyRxEidetcetDis	RW 0x0	PIPE Interface RX Electrical Idle (or Squelch) Detect Disable 0 = Enable: Enables squelch detector. 1 = Disable: Disables squelch detector.
3:2	MacPhyPwDn0	RW 0x0	PIPE Interface Power State of the Transceiver **NOTE:** When the PHY is operating with the PCIe or USB MAC, this signal is driven by the MAC. 0 = P0 normal operation 1 = P1 power saving state 2 = P2 power saving state 3 = P3 lowest power saving state
1	RxDataFlip	RW 0x0	Flip PHY RX Data 0 = No Flip 1 = Flip
0	TxDataFlip	RW 0x0	Flip PHY TX Data 0 = No Flip 1 = Flip

Table 1472: Common PHY<n> Configuration 4 Register (n=0–5)
Offset: PHY_NUM0: 0x0001830C, PHY_NUM1: 0x00018334, PHY_NUM2: 0x0001835C, PHY_NUM3: 0x00018384, PHY_NUM4: 0x000183AC, PHY_NUM5: 0x000183D4
Offset Formula: 0x0001830C+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31	MAC PHY Eq In Prog	RW 0x0	Link equalization procedure is in progress.
30	MAC PHY Tx Start Block	RW 0x0	Only used at the 8.0 GT/s signaling rate. The MAC tells the PHY the starting byte for a 128b block. The starting byte for a 128b block must always start with bit 0 of the data interface.
29:28	MAC PHY Tx Sync Header	RW 0x0	Provides the sync header for the PHY to use in the next 130b block. The PHY reads this value when the TxStartBlock signal is asserted.
27:24	MAC PHY Tx Data Skip	RW 0x0	Only used at the 8.0 GT/s signaling rate. The MAC instructs the PHY to ignore one or more bytes on the data interface. A value of 0 indicates the PHY will use the byte. A value of 1 indicates the PHY will not use the byte.

Copyright © 2015 Marvell

Table 1472: Common PHY<n> Configuration 4 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x0001830C, PHY_NUM1: 0x00018334, PHY_NUM2: 0x0001835C, PHY_NUM3: 0x00018384, PHY_NUM4: 0x000183AC, PHY_NUM5: 0x000183D4
Offset Formula: 0x0001830C+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
23	Pin Tx Train Enable	RW 0x0	When asserted, indicates that the local PHY will begin its transmitter adaptation processes. When de-asserted, before the assertion of <Tx Train Complete>, indicates the local PHY will cease any transmitter adaptation processes currently in progress.
22	Pin Ssc En	RW 0x0	Spread Spectrum Clock Enable. 0 = Disable: Disables SSC. 1 = Enable: Enables SSC.
21:4	Reserved	RSVD 0x0	Reserved
3	Pin Dfe Update Dis	RW 0x1	DFE Update Disable. This pin is valid when register dfe_control_from_pin is set to high. 0 = Enable: Enable DFE adaptation. 1 = Disable: Disable DFE adaptation but keep DFE current coefficient values.
2	Pin Dfe Pat Dis	RW 0x1	DFE Pattern Protection Disable. This pin is valid when register dfe_control_from_pin is set to high. 0 = Enable: Enables the DFE Pattern Protection function. 1 = Disable: Disables the DFE Pattern Protection function.
1	Pin Dfe En	RW 0x1	DFE Function Enable. This pin is valid when register dfe_control_from_pin is set to high. 0 = Disable: Disables the DFE function and sets F1-F5 to 0. 1 = Enable: Enables the DFE function.
0	Pin Isolation Dis	RW 0x1	Isolation Digital Power Supply Disable This signal controls the PHY digital power supply switch between DVDD and DVDD_ISOLATION. It should be tied to 1 if there is only one digital power supply for the PHY. 0 = Power Saving: Power saving mode in which DVDD can be turned off. 1 = Normal Function: Normal function mode in which DVDD must be on.

Table 1473: Common PHY<n> Configuration 5 Register (n=0–5)
Offset: PHY_NUM0: 0x00018310, PHY_NUM1: 0x00018338, PHY_NUM2: 0x00018360, PHY_NUM3: 0x00018388, PHY_NUM4: 0x000183B0, PHY_NUM5: 0x000183D8
Offset Formula: 0x00018310+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:29	Reserved	RSVD 0x0	Reserved
28	MAC PHY EQ FOM DIRN	RW 0x0	Indicates either use of Figure of Merit (FOM) mode or Direction Change (DIR) mode when the MAC requests the PHY to perform evaluation feedback by asserting the MAC_PHY_EQ_REQ_REMOTE signal (in PIPE v4 mode). 0 = DIR 1 = FOM

Copyright © 2015 Marvell

Table 1473: Common PHY<n> Configuration 5 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018310, PHY_NUM1: 0x00018338, PHY_NUM2: 0x00018360, PHY_NUM3: 0x00018388, PHY_NUM4: 0x000183B0, PHY_NUM5: 0x000183D8
Offset Formula: 0x00018310+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
27	MAC PHY TX Data Valid	RW 0x1	This signal allows the MAC to instruct the PHY to ignore the data interface for one clock cycle. A value of 1 indicates the PHY will use the data. A value of 0 indicates the PHY will not use the data.
26	MAC PHY Block Align	RW 0x0	Controls whether the PHY performs block alignment. This signals is only used at the 8.0 GT/s signaling rate. This signal is enabled by the PIPE register CFG_BLK_ALIGN_CTRL[1].
25:23	MAC PHY Eq Preset High	RW 0x0	Local transmitter equalization preset value. Receiver preset hint.
22:21	Reserved	RSVD 0x0	Reserved
20	Pin Sq Oob Sel	RW 0x0	Squelch Detector Selection for out-of-band (OOB) . This signal selects the analog squelch detector for OOB signal detection. It should be set to 1 during OOB detection and cleared to 0 after OOB detection, which should be controlled by the SoC state machine. This control is only applied to SATA or SAS applications. For all the other applications, it should be cleared to 0. 0 = Disable: Selects non-OOB squelch detector. 1 = Enable: Selects OOB squelch detector.
19	Pin Rx Train Enable	RW 0x0	When asserted, indicates the local PHY will begin its receiver adaptation processes. This is utilized during the existing SAS 2.0 receiver training states and will not be utilized during the new SAS 3.0 transmitter training states. **NOTE:** In the current SAS 3.0 proposals, the existing SAS 2.0 receiver training states (Train-SNW) are still visited following completion of the existing SAS 3.0 training states. When de-asserted before the assertion of <Rx Train Complete>, indicates the local PHY will cease any receiver adaptation processes currently in progress.
18:14	Lane Num	RW <%n>	Lane Number For x1 lane, tie to 0. For multi-lanes, assign the lane number. When PIN_PADDR[13:9] == LANE_NUM[4:0], the lane is enabled to be accessed by APB3. Otherwise, APB3 accessing is disabled.
13	Pin Dev Sel	RW 0x0	Device Select for Serial Interface (SIF) For x1 lane, tie to 1. For multi-lanes, map to the SoC register bits or to GPIO pads. 0 = Disable: Serial interface is disabled for current lane. 1 = Enable: Serial interface is enabled for current lane.
12	MAC PHY Eq Coeff Rejected	RW 0x0	The previous set of remote coefficients were rejected.
11	MAC PHY Eq Req Remote	RW 0x0	Request a new set of equalization coefficient values or preset values for remote transmitter.

Copyright © 2015 Marvell

Table 1473: Common PHY<n> Configuration 5 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018310, PHY_NUM1: 0x00018338, PHY_NUM2: 0x00018360, PHY_NUM3: 0x00018388, PHY_NUM4: 0x000183B0, PHY_NUM5: 0x000183D8
Offset Formula: 0x00018310+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
10	MAC PHY Eq Req Local	RW 0x0	Request equalization coefficient values from the local PHY.
9	MAC PHY Eq Coeff Valid	RW 0x0	MAC PHY equalization coefficient valid.
8:5	MAC PHY Eq Preset	RW 0x0	Local transmitter equalization preset value. Transmitter preset.
4	MAC PHY Eq Preset Valid	RW 0x0	Local transmitter equalization preset valid.
3	MAC PHY Cmp Preset Valid	RW 0x0	When asserted, the compliance preset data is valid at the MAC_PHY_EQ_PRESET[3:0] bus.
2	MAC PHY Lb In Prog	RW 0x0	Link's LTSSM is in loopback state.
1:0	MAC PHY Eq Phase	RW 0x0	The current phase of equalization procedure. 0 = Phase 0 1 = Phase 1 2 = Phase 2 3 = Phase 3

Table 1474: Common PHY<n> Configuration 6 Register (n=0–5)
Offset: PHY_NUM0: 0x00018314, PHY_NUM1: 0x0001833C, PHY_NUM2: 0x00018364, PHY_NUM3: 0x0001838C, PHY_NUM4: 0x000183B4, PHY_NUM5: 0x000183DC
Offset Formula: 0x00018314+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:18	Reserved	RSVD 0x0	Reserved
17:0	MAC PHY Eq Coeff	RW 0x0	Local transmitter equalization coefficients.

Table 1475: Common PHY<n> Status 1 Register (n=0–5)
Offset: PHY_NUM0: 0x00018318, PHY_NUM1: 0x00018340, PHY_NUM2: 0x00018368, PHY_NUM3: 0x00018390, PHY_NUM4: 0x000183B8, PHY_NUM5: 0x000183E0
Offset Formula: 0x00018318+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:27	Reserved	RSVD 0x0	Reserved
26	RefClkDisAck	RO 0x0	Acknowledge signal for RefClkDis field of the Common PHY<n> Configuration 1 register. The reference clock can be disabled only when the <RefClkDisAck> and <RefClkDis> fields are both set to 1.

Copyright © 2015 Marvell

Table 1475: Common PHY<n> Status 1 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018318, PHY_NUM1: 0x00018340, PHY_NUM2: 0x00018368, PHY_NUM3: 0x00018390, PHY_NUM4: 0x000183B8, PHY_NUM5: 0x000183E0
Offset Formula: 0x00018318+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
25:10	Reserved	RSVD 0x0	Reserved
9	Sq Detected	RO 0x0	Squelch Detector Output This signal is used for out-of-band (OOB) signal detection. The SoC must be synchronized and glitch-free before its use. **NOTE:** This signal is also sampled by MACs that support it. 0 = Signal detected. 1 = Signal not detected.
8:6	PHY MAC Rx Status	RO 0x0	Encodes receiver status and error codes. See specifications for encoding table. **NOTE:** This signal is also sampled by MACs that support it.
5	PHY MAC Rx Elec Idle	RO 0x0	Indicates Detection of an Electrical Idle. There are two definition for this signal. The first definition is as follows: 0 = LFPS is detected. 1 = No LPFS is detected. The second definition is as follows: **NOTE:** This signal is also sampled by MACs that support it. 0 = LPFS or active data is detected. 1 = Electrical idle is detected.
4	PHY MAC Power Present	RO 0x0	Vbus Present This signal is used in USB SuperSpeed mode and indicates the presence of the Vbus. NOTES: 1. This signal is also sampled by MACs that support it. 2. This signal is reserved for future implementation only.
3	PLL Ready Tx	RO 0x0	PLL ready for TX. This field indicates that the PHY TX clock is ready and the TX can send out data. **NOTE:** This signal is also sampled by MACs that support it.
2	PLL Ready Rx	RO 0x0	PLL ready for RX. This field indicates that the PHY RX clock is ready and the PIN_RX_INIT can be asserted. **NOTE:** This signal is also sampled by MACs that support it.

Copyright © 2015 Marvell

Table 1475: Common PHY<n> Status 1 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018318, PHY_NUM1: 0x00018340, PHY_NUM2: 0x00018368, PHY_NUM3: 0x00018390, PHY_NUM4: 0x000183B8, PHY_NUM5: 0x000183E0
Offset Formula: 0x00018318+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
1	PHY MAC PHY Status	RO 0x0	PHY Status. This signal communicates the completion of several PHY functions, including power management state transitions, rate changes and receiver detection. This signal is generated by the PCLK in P0, P0S, and P1 state. This signal is generated by the SCLK in the P2 state. **NOTE:** This signal is also sampled by MACs that support it.
0	RX Init Done	RO 0x0	Receiver Initialization Done. This is a level signal. It indicates that the PHY DTL and DFE have entered optimized states.

Table 1476: Common PHY<n> Status 2 Register (n=0–5)
Offset: PHY_NUM0: 0x0001831C, PHY_NUM1: 0x00018344, PHY_NUM2: 0x0001836C, PHY_NUM3: 0x00018394, PHY_NUM4: 0x000183BC, PHY_NUM5: 0x000183E4
Offset Formula: 0x0001831C+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31	PHY MAC Rx Start Block	RO 0x0	Only used at the 8.0 GT/s signaling rate. The PHY instructs the MAC of the starting byte for a 128b block. The starting byte for a 128b block must always start with bit 0 of the data interface.
30	Pin Tx Train Failed	RO 0x0	Valid when tx_train_complete is asserted. Indicates the local PHY has encountered a problem during training or could not converge.
29	Pin Tx Train Complete	RO 0x0	When asserted, indicates the local PHY's transmitter training process has completed. This signal will go to high when TX training is complete and it will go to low when PIN_TX_TRAIN_ENABLE is low.
28	Pin Remote Ctrl Field Ready	RO 0x0	When asserted, indicates the local PHY is ready to receive TX coefficient update requests from the remote PHY.
27	Pin Rx Train Failed	RO 0x0	Valid when rx_train_complete is asserted. Indicates the local PHY has encountered a problem during training or could not converge.
26	Pin Rx Train Complete	RO 0x0	When asserted, indicates the local PHY has completed its receiver adaptation process. This signal will go to high when RX training is complete, that is PIN_RX_TRAIN_ENABLE is low.
25	Pin Local Status Field Valid	RO 0x0	When asserted, indicates that new values of the Local TX status on PIN_LOCAL_STATUS_FIELD[5:0] are valid.
24:19	Pin Local Status Field	RO 0x0	Local TX Status Field bits for Remote device. Indicates Local TX coefficients are max, min, update complete, or ready.

Copyright © 2015 Marvell

Table 1476: Common PHY<n> Status 2 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x0001831C, PHY_NUM1: 0x00018344, PHY_NUM2: 0x0001836C, PHY_NUM3: 0x00018394, PHY_NUM4: 0x000183BC, PHY_NUM5: 0x000183E4
Offset Formula: 0x0001831C+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
18	Pin Local Ctrl Field Valid	RO 0x0	When asserted, indicates that new values output from the local PHY on PIN_LOCAL_CTRL_FIELD[5:0] and PIN_LOCAL_CTRL_FIELD_RESET are valid.
17:14	Pin Local Ctrl Field Reset	RO 0x0	Valid when Dev_Ctrl_Field_Valid is asserted. Indicates a request to the remote PHY to reset its TX coefficients.
13:8	Pin Local Ctrl Field	RO 0x0	Device Control Field bits for TX host coefficient updates. Requests for decrement, increment, or hold.
7:0	PHY Reserved Out	RO 0x0	PhyReservedOut[7:2,0]: Reserved output pin for USB3 PIPE. PhyReservedOut[1]/PHY_RCB_RX_EN: Reference clock buffer receiver enable.

Table 1477: Common PHY<n> Status 3 Register (n=0–5)
Offset: PHY_NUM0: 0x00018320, PHY_NUM1: 0x00018348, PHY_NUM2: 0x00018370, PHY_NUM3: 0x00018398, PHY_NUM4: 0x000183C0, PHY_NUM5: 0x000183E8
Offset Formula: 0x00018320+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:24	PHY MAC Eq Lf	RO 0x0	Low Frequency (LF) Parameter from Local PHY
23:18	PHY MAC Eq Fs	RO 0x0	Full Swing (FS) Parameter from Local PHY
17	PHY MAC Eq Bypass Phase_23	RO 0x0	Request LTSSM to bypass phase 2 and 3 of the link equalization procedure.
16	PHY MAC Eq Reset EIEOS Count	RO 0x0	Reset EIEOS counter during training.
15	PHY MAC Eq Complete	RO 0x0	Equalization is complete and the parameters have reached their optimal settings.
14	PHY MAC Eq Coeff Req Ack	RO 0x0	Remote Coefficient/Preset Valid
13	PHY MAC Eq Coeff Valid	RO 0x0	Remote Transmitter Equalization Coefficient Valid
12:9	PHY MAC Eq Preset	RO 0x0	Remote Transmitter Equalization Preset Value
8	PHY MAC Eq Preset Valid	RO 0x0	Remote Transmitter Equalization Preset Valid

Copyright © 2015 Marvell

Table 1477: Common PHY<n> Status 3 Register (n=0–5) (Continued)
Offset: PHY_NUM0: 0x00018320, PHY_NUM1: 0x00018348, PHY_NUM2: 0x00018370, PHY_NUM3: 0x00018398, PHY_NUM4: 0x000183C0, PHY_NUM5: 0x000183E8
Offset Formula: 0x00018320+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
7:6	PHY MAC Eq Accept Status	RO 0x0	Accept status of local transmitter equalization coefficient. Other values are Reserved 1 = Accepted: Coefficients are valid – accepted. 2 = Rejected: Coefficients are invalid – rejected
5:4	PHY MAC Rx Sync Header	RO 0x0	Provides the sync header for the MAC to use in the next 130b block. The MAC reads this value when the RxStartBlock signal is asserted.
3:0	PHY MAC Rx Data Skip	RO 0x0	Only used at the 8.0 GT/s signaling rate. The PHY instructs the MAC to ignore one or more bytes on the data interface. A value of 0 indicates the PHY will use the byte A value of 1 indicates the PHY will not use the byte.

Table 1478: Common PHY<n> Status 4 Register (n=0–5)
Offset: PHY_NUM0: 0x00018324, PHY_NUM1: 0x0001834C, PHY_NUM2: 0x00018374, PHY_NUM3: 0x0001839C, PHY_NUM4: 0x000183C4, PHY_NUM5: 0x000183EC
Offset Formula: 0x00018324+n*0x28: where n (0-5) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:18	Reserved	RSVD 0x0	Reserved
17:0	PHY MAC Eq Coeff	RO 0x0	Remote transmitter equalization coefficients. When the rate is 2.5 or 5.0 GT/s: Don't care. When the rate is 8.0 GT/s: Bit[5:0]: C-1 Bit[11:6]: C 0 Bit[17:12]: C+1

Table 1479: Common PHYs Selectors Register
Offset: 0x000183FC

Bit	Field	Type/InitVal	Description
NOTE: For the fields in this register, do not use any values that are not shown in the field description.			
31	PCIe x4 Enable	RW 0x0	Essential for x4 routing (clocks, COMPHYs interfaces): 0 = PCIe0 x1: PCIe0 is used as x1. 1 = PCIe0 x4: PCIe0 is used as x4.
30:28	Reserved	RSVD 0x0	Reserved
27:24	COMPHY6 Selector	RW 0x0	COMPHY6 Selector 0 = Unconnected 1 = PCIe Port1 3 = Reserved 12 = GbE v3 Port0

Copyright © 2015 Marvell

Table 1479: Common PHYs Selectors Register (Continued)
Offset: 0x000183FC

Bit	Field	Type/InitVal	Description
23:20	COMPHY5 Selector	RW 0x0	COMPHY5 Selector 0 = Unconnected 1 = PCIe Port2 2 = SATA3 Port2 3 = GbE Port2 4 = USB3 Host Port1 5 = USB3 Device 7 = Reserved
19:16	COMPHY4 Selector	RW 0x0	COMPHY4 Selector 0 = Unconnected 1 = PCIe Port1 2 = Unconnected1 3 = GbE Port1 4 = USB3 Host Port0 5 = USB3 Device 6 = SATA3 Port2 7 = PCIe Port2
15:12	COMPHY3 Selector	RW 0x0	COMPHY3 Selector 0 = Unconnected 1 = PCIe Port3 2 = PCIe Port03 3 = SATA3 Port3 4 = GbE Port2 5 = USB3 Host Port1 6 = USB3 Device
11:8	COMPHY2 Selector	RW 0x0	COMPHY2 Selector 0 = Unconnected 1 = PCIe Port1 2 = PCIe Port02 3 = SATA3 Port1 4 = GbE Port1
7:4	COMPHY1 Selector	RW 0x0	COMPHY1 Selector 0 = Unconnected 1 = PCIe Port00 2 = PCIe Port01 3 = SATA3 Port0 4 = GbE Port0 5 = GbE Port1 6 = USB3_Host Port0 7 = QSGMII 10 = Unconnected1
3:0	COMPHY0 Selector	RW 0x0	COMPHY0 Selector 0 = Unconnected 1 = PCIe Port00 2 = SATA3 Port0 3 = GbE Port0

Copyright © 2015 Marvell

A.24 Common PHY and PIPE Interface Registers

The following table provides a summarized list of all registers that belong to the Common PHY and PIPE Interface, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers

Register Name	Offset	Table and Page
Common PHY and PIPE Interface		
Chip ID Register (n=0–6)	Port0: 0x000A0000, Port1: 0x000A0800, Port2: 0x000A1000, Port3: 0x000A1800, Port4: 0x000A2000, Port5: 0x000A2800,	Table 1481, p. 1204
Power and PLL Control Register (n=0–6)	Port0: 0x000A0004, Port1: 0x000A0804, Port2: 0x000A1004, Port3: 0x000A1804, Port4: 0x000A2004, Port5: 0x000A2804,	Table 1482, p. 1205
KVCO Calibration Control Register (n=0–6)	Port0: 0x000A0008, Port1: 0x000A0808, Port2: 0x000A1008, Port3: 0x000A1808, Port4: 0x000A2008, Port5: 0x000A2808,	Table 1483, p. 1208
Impedance Calibration Control Register (n=0–6)	Port0: 0x000A000C, Port1: 0x000A080C, Port2: 0x000A100C, Port3: 0x000A180C, Port4: 0x000A200C, Port5: 0x000A280C,	Table 1484, p. 1210
Impedance and TX SSC Control Register (n=0–6)	Port0: 0x000A0010, Port1: 0x000A0810, Port2: 0x000A1010, Port3: 0x000A1810, Port4: 0x000A2010, Port5: 0x000A2810,	Table 1485, p. 1210
Transmitter Frequency Offset Register (n=0–6)	Port0: 0x000A0014, Port1: 0x000A0814, Port2: 0x000A1014, Port3: 0x000A1814, Port4: 0x000A2014, Port5: 0x000A2814,	Table 1486, p. 1211
Squelch and FFE Setting Register (n=0–6)	Port0: 0x000A0018, Port1: 0x000A0818, Port2: 0x000A1018, Port3: 0x000A1818, Port4: 0x000A2018, Port5: 0x000A2818,	Table 1487, p. 1211

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
DFE Resolution Control Register (n=0–6)	Port0: 0x000A001C, Port1: 0x000A081C, Port2: 0x000A101C, Port3: 0x000A181C, Port4: 0x000A201C, Port5: 0x000A281C,	Table 1488, p. 1212
DFE Update Control Register (n=0–6)	Port0: 0x000A0020, Port1: 0x000A0820, Port2: 0x000A1020, Port3: 0x000A1820, Port4: 0x000A2020, Port5: 0x000A2820,	Table 1489, p. 1213
DFE F0-F2 Coefficient Control Register (n=0–6)	Port0: 0x000A0024, Port1: 0x000A0824, Port2: 0x000A1024, Port3: 0x000A1824, Port4: 0x000A2024, Port5: 0x000A2824,	Table 1490, p. 1214
DFE F3-F5 Coefficient Control Register (n=0–6)	Port0: 0x000A0028, Port1: 0x000A0828, Port2: 0x000A1028, Port3: 0x000A1828, Port4: 0x000A2028, Port5: 0x000A2828,	Table 1491, p. 1214
Adapted DFE Coefficient 0 Register (n=0–6)	Port0: 0x000A002C, Port1: 0x000A082C, Port2: 0x000A102C, Port3: 0x000A182C, Port4: 0x000A202C, Port5: 0x000A282C,	Table 1492, p. 1215
Adapted DFE Coefficient 1 Register (n=0–6)	Port0: 0x000A0030, Port1: 0x000A0830, Port2: 0x000A1030, Port3: 0x000A1830, Port4: 0x000A2030, Port5: 0x000A2830,	Table 1493, p. 1215
Generation 1 Setting 0 Register (n=0–6)	Port0: 0x000A0034, Port1: 0x000A0834, Port2: 0x000A1034, Port3: 0x000A1834, Port4: 0x000A2034, Port5: 0x000A2834,	Table 1494, p. 1216
Generation 1 Setting 1 Register (n=0–6)	Port0: 0x000A0038, Port1: 0x000A0838, Port2: 0x000A1038, Port3: 0x000A1838, Port4: 0x000A2038, Port5: 0x000A2838,	Table 1495, p. 1216
Generation 2 Setting 0 Register (n=0–6)	Port0: 0x000A003C, Port1: 0x000A083C, Port2: 0x000A103C, Port3: 0x000A183C, Port4: 0x000A203C, Port5: 0x000A283C,	Table 1496, p. 1218

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Generation 2 Setting 1 Register (n=0–6)	Port0: 0x000A0040, Port1: 0x000A0840, Port2: 0x000A1040, Port3: 0x000A1840, Port4: 0x000A2040, Port5: 0x000A2840,	Table 1497, p. 1219
Generation 3 Setting 0 Register (n=0–6)	Port0: 0x000A0044, Port1: 0x000A0844, Port2: 0x000A1044, Port3: 0x000A1844, Port4: 0x000A2044, Port5: 0x000A2844,	Table 1498, p. 1220
Generation 3 Setting 1 Register (n=0–6)	Port0: 0x000A0048, Port1: 0x000A0848, Port2: 0x000A1048, Port3: 0x000A1848, Port4: 0x000A2048, Port5: 0x000A2848,	Table 1499, p. 1221
Generation 4 Setting 0 Register (n=0–6)	Port0: 0x000A004C, Port1: 0x000A084C, Port2: 0x000A104C, Port3: 0x000A184C, Port4: 0x000A204C, Port5: 0x000A284C,	Table 1500, p. 1223
Generation 4 Setting 1 Register (n=0–6)	Port0: 0x000A0050, Port1: 0x000A0850, Port2: 0x000A1050, Port3: 0x000A1850, Port4: 0x000A2050, Port5: 0x000A2850,	Table 1501, p. 1223
PHY Test Control 0 Register (n=0–6)	Port0: 0x000A0054, Port1: 0x000A0854, Port2: 0x000A1054, Port3: 0x000A1854, Port4: 0x000A2054, Port5: 0x000A2854,	Table 1502, p. 1225
PHY Test Data 0 Register (n=0–6)	Port0: 0x000A0058, Port1: 0x000A0858, Port2: 0x000A1058, Port3: 0x000A1858, Port4: 0x000A2058, Port5: 0x000A2858,	Table 1503, p. 1226
PHY Test Data 1 Register (n=0–6)	Port0: 0x000A005C, Port1: 0x000A085C, Port2: 0x000A105C, Port3: 0x000A185C, Port4: 0x000A205C, Port5: 0x000A285C,	Table 1504, p. 1227
PHY Test Data 2 Register (n=0–6)	Port0: 0x000A0060, Port1: 0x000A0860, Port2: 0x000A1060, Port3: 0x000A1860, Port4: 0x000A2060, Port5: 0x000A2860,	Table 1505, p. 1227

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
PHY Test Data 3 Register (n=0–6)	Port0: 0x000A0064, Port1: 0x000A0864, Port2: 0x000A1064, Port3: 0x000A1864, Port4: 0x000A2064, Port5: 0x000A2864,	Table 1506, p. 1228
PHY Test Data 4 Register (n=0–6)	Port0: 0x000A0068, Port1: 0x000A0868, Port2: 0x000A1068, Port3: 0x000A1868, Port4: 0x000A2068, Port5: 0x000A2868,	Table 1507, p. 1228
PHY Test Data 5 Register (n=0–6)	Port0: 0x000A006C, Port1: 0x000A086C, Port2: 0x000A106C, Port3: 0x000A186C, Port4: 0x000A206C, Port5: 0x000A286C,	Table 1508, p. 1229
PHY Test PRBS Counter 0 Register (n=0–6)	Port0: 0x000A0070, Port1: 0x000A0870, Port2: 0x000A1070, Port3: 0x000A1870, Port4: 0x000A2070, Port5: 0x000A2870,	Table 1509, p. 1229
PHY Test PRBS Counter 1 Register (n=0–6)	Port0: 0x000A0074, Port1: 0x000A0874, Port2: 0x000A1074, Port3: 0x000A1874, Port4: 0x000A2074, Port5: 0x000A2874,	Table 1510, p. 1229
PHY Test PRBS Counter 2 Register (n=0–6)	Port0: 0x000A0078, Port1: 0x000A0878, Port2: 0x000A1078, Port3: 0x000A1878, Port4: 0x000A2078, Port5: 0x000A2878,	Table 1511, p. 1229
PHY Test PRBS Error Counter 0 Register (n=0–6)	Port0: 0x000A007C, Port1: 0x000A087C, Port2: 0x000A107C, Port3: 0x000A187C, Port4: 0x000A207C, Port5: 0x000A287C,	Table 1512, p. 1230
PHY Test PRBS Error Counter 1 Register (n=0–6)	Port0: 0x000A0080, Port1: 0x000A0880, Port2: 0x000A1080, Port3: 0x000A1880, Port4: 0x000A2080, Port5: 0x000A2880,	Table 1513, p. 1230
PHY Test OOB 0 Register (n=0–6)	Port0: 0x000A0084, Port1: 0x000A0884, Port2: 0x000A1084, Port3: 0x000A1884, Port4: 0x000A2084, Port5: 0x000A2884,	Table 1514, p. 1230

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
PHY Test OOB Register (n=0–6)	Port0: 0x000A0088, Port1: 0x000A0888, Port2: 0x000A1088, Port3: 0x000A1888, Port4: 0x000A2088, Port5: 0x000A2888,	Table 1515, p. 1230
Digital Loopback Enable Register (n=0–6)	Port0: 0x000A008C, Port1: 0x000A088C, Port2: 0x000A108C, Port3: 0x000A188C, Port4: 0x000A208C, Port5: 0x000A288C,	Table 1516, p. 1231
Sync Pattern Register (n=0–6)	Port0: 0x000A0090, Port1: 0x000A0890, Port2: 0x000A1090, Port3: 0x000A1890, Port4: 0x000A2090, Port5: 0x000A2890,	Table 1517, p. 1232
Sync Mask and Maximal Generation Setting Register (n=0–6)	Port0: 0x000A0094, Port1: 0x000A0894, Port2: 0x000A1094, Port3: 0x000A1894, Port4: 0x000A2094, Port5: 0x000A2894,	Table 1518, p. 1233
PHY Isolation Mode Control Register (n=0–6)	Port0: 0x000A0098, Port1: 0x000A0898, Port2: 0x000A1098, Port3: 0x000A1898, Port4: 0x000A2098, Port5: 0x000A2898,	Table 1519, p. 1234
EOM Circuit Control Register (n=0–6)	Port0: 0x000A009C, Port1: 0x000A089C, Port2: 0x000A109C, Port3: 0x000A189C, Port4: 0x000A209C, Port5: 0x000A289C,	Table 1520, p. 1236
EOM Set 1 Control 0 Register (n=0–6)	Port0: 0x000A00A0, Port1: 0x000A08A0, Port2: 0x000A10A0, Port3: 0x000A18A0, Port4: 0x000A20A0, Port5: 0x000A28A0,	Table 1521, p. 1237
EOM Set 1 Control 1 Register (n=0–6)	Port0: 0x000A00A4, Port1: 0x000A08A4, Port2: 0x000A10A4, Port3: 0x000A18A4, Port4: 0x000A20A4, Port5: 0x000A28A4,	Table 1522, p. 1239
EOM Set 1 Population Counter 0 Register (n=0–6)	Port0: 0x000A00A8, Port1: 0x000A08A8, Port2: 0x000A10A8, Port3: 0x000A18A8, Port4: 0x000A20A8, Port5: 0x000A28A8,	Table 1523, p. 1239

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
EOM Set 1 Population Counter 1 Register (n=0–6)	Port0: 0x000A00AC, Port1: 0x000A08AC, Port2: 0x000A10AC, Port3: 0x000A18AC, Port4: 0x000A20AC, Port5: 0x000A28AC,	Table 1524, p. 1240
EOM Set 1 Population Counter 2 Register (n=0–6)	Port0: 0x000A00B0, Port1: 0x000A08B0, Port2: 0x000A10B0, Port3: 0x000A18B0, Port4: 0x000A20B0, Port5: 0x000A28B0,	Table 1525, p. 1240
EOM Set 1 Population Counter 3 Register (n=0–6)	Port0: 0x000A00B4, Port1: 0x000A08B4, Port2: 0x000A10B4, Port3: 0x000A18B4, Port4: 0x000A20B4, Port5: 0x000A28B4,	Table 1526, p. 1240
EOM Set 1 Threshold Counter 0 Register (n=0–6)	Port0: 0x000A00B8, Port1: 0x000A08B8, Port2: 0x000A10B8, Port3: 0x000A18B8, Port4: 0x000A20B8, Port5: 0x000A28B8,	Table 1527, p. 1240
EOM Set 1 Threshold Counter 1 Register (n=0–6)	Port0: 0x000A00BC, Port1: 0x000A08BC, Port2: 0x000A10BC, Port3: 0x000A18BC, Port4: 0x000A20BC, Port5: 0x000A28BC,	Table 1528, p. 1241
EOM Set 1 Threshold Counter 2 Register (n=0–6)	Port0: 0x000A00C0, Port1: 0x000A08C0, Port2: 0x000A10C0, Port3: 0x000A18C0, Port4: 0x000A20C0, Port5: 0x000A28C0,	Table 1529, p. 1241
EOM Set 1 Threshold Counter 3 Register (n=0–6)	Port0: 0x000A00C4, Port1: 0x000A08C4, Port2: 0x000A10C4, Port3: 0x000A18C4, Port4: 0x000A20C4, Port5: 0x000A28C4,	Table 1530, p. 1242
EOM Set 1 Sample Counter 0 Register (n=0–6)	Port0: 0x000A00C8, Port1: 0x000A08C8, Port2: 0x000A10C8, Port3: 0x000A18C8, Port4: 0x000A20C8, Port5: 0x000A28C8,	Table 1531, p. 1242
EOM Set 1 Sample Counter 1 Register (n=0–6)	Port0: 0x000A00CC, Port1: 0x000A08CC, Port2: 0x000A10CC, Port3: 0x000A18CC, Port4: 0x000A20CC, Port5: 0x000A28CC,	Table 1532, p. 1242

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
EOM Set 1 Sample Counter 2 Register (n=0–6)	Port0: 0x000A00D0, Port1: 0x000A08D0, Port2: 0x000A10D0, Port3: 0x000A18D0, Port4: 0x000A20D0, Port5: 0x000A28D0,	Table 1533, p. 1242
EOM Set 1 Sample Counter 3 Register (n=0–6)	Port0: 0x000A00D4, Port1: 0x000A08D4, Port2: 0x000A10D4, Port3: 0x000A18D4, Port4: 0x000A20D4, Port5: 0x000A28D4,	Table 1534, p. 1243
EOM Set 1 Error Counter 0 Register (n=0–6)	Port0: 0x000A00D8, Port1: 0x000A08D8, Port2: 0x000A10D8, Port3: 0x000A18D8, Port4: 0x000A20D8, Port5: 0x000A28D8,	Table 1535, p. 1243
EOM Set 1 Error Counter 1 Register (n=0–6)	Port0: 0x000A00DC, Port1: 0x000A08DC, Port2: 0x000A10DC, Port3: 0x000A18DC, Port4: 0x000A20DC, Port5: 0x000A28DC,	Table 1536, p. 1243
EOM Set 1 Error Counter 2 Register (n=0–6)	Port0: 0x000A00E0, Port1: 0x000A08E0, Port2: 0x000A10E0, Port3: 0x000A18E0, Port4: 0x000A20E0, Port5: 0x000A28E0,	Table 1537, p. 1243
EOM Set 1 Error Counter 3 Register (n=0–6)	Port0: 0x000A00E4, Port1: 0x000A08E4, Port2: 0x000A10E4, Port3: 0x000A18E4, Port4: 0x000A20E4, Port5: 0x000A28E4,	Table 1538, p. 1244
External Digital Reserved 0 Register (n=0–6)	Port0: 0x000A00E8, Port1: 0x000A08E8, Port2: 0x000A10E8, Port3: 0x000A18E8, Port4: 0x000A20E8, Port5: 0x000A28E8,	Table 1539, p. 1244
DFE Step Size 0 Register (n=0–6)	Port0: 0x000A00EC, Port1: 0x000A08EC, Port2: 0x000A10EC, Port3: 0x000A18EC, Port4: 0x000A20EC, Port5: 0x000A28EC,	Table 1540, p. 1244
DFE Step Size 1 Register (n=0–6)	Port0: 0x000A00F0, Port1: 0x000A08F0, Port2: 0x000A10F0, Port3: 0x000A18F0, Port4: 0x000A20F0, Port5: 0x000A28F0,	Table 1541, p. 1246

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Generation 1 Setting 2 Register (n=0–6)	Port0: 0x000A00F4, Port1: 0x000A08F4, Port2: 0x000A10F4, Port3: 0x000A18F4, Port4: 0x000A20F4, Port5: 0x000A28F4,	Table 1542, p. 1247
Generation 2 Setting 2 Register (n=0–6)	Port0: 0x000A00F8, Port1: 0x000A08F8, Port2: 0x000A10F8, Port3: 0x000A18F8, Port4: 0x000A20F8, Port5: 0x000A28F8,	Table 1543, p. 1248
Generation 3 Setting 2 Register (n=0–6)	Port0: 0x000A00FC, Port1: 0x000A08FC, Port2: 0x000A10FC, Port3: 0x000A18FC, Port4: 0x000A20FC, Port5: 0x000A28FC,	Table 1544, p. 1249
Generation 4 Setting 2 Register (n=0–6)	Port0: 0x000A0100, Port1: 0x000A0900, Port2: 0x000A1100, Port3: 0x000A1900, Port4: 0x000A2100, Port5: 0x000A2900,	Table 1545, p. 1249
VTHIMPCAL Value Control Register (n=0–6)	Port0: 0x000A0104, Port1: 0x000A0904, Port2: 0x000A1104, Port3: 0x000A1904, Port4: 0x000A2104, Port5: 0x000A2904,	Table 1546, p. 1250
VDD Calibration Control 0 Register (n=0–6)	Port0: 0x000A0108, Port1: 0x000A0908, Port2: 0x000A1108, Port3: 0x000A1908, Port4: 0x000A2108, Port5: 0x000A2908,	Table 1547, p. 1251
VDD Calibration Control 1 Register (n=0–6)	Port0: 0x000A010C, Port1: 0x000A090C, Port2: 0x000A110C, Port3: 0x000A190C, Port4: 0x000A210C, Port5: 0x000A290C,	Table 1548, p. 1251
VDD Calibration Control 2 Register (n=0–6)	Port0: 0x000A0110, Port1: 0x000A0910, Port2: 0x000A1110, Port3: 0x000A1910, Port4: 0x000A2110, Port5: 0x000A2910,	Table 1549, p. 1251
VDD Calibration Control 3 Register (n=0–6)	Port0: 0x000A0114, Port1: 0x000A0914, Port2: 0x000A1114, Port3: 0x000A1914, Port4: 0x000A2114, Port5: 0x000A2914,	Table 1550, p. 1251

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Analog Reserved 1 Register (n=0–6)	Port0: 0x000A0118, Port1: 0x000A0918, Port2: 0x000A1118, Port3: 0x000A1918, Port4: 0x000A2118, Port5: 0x000A2918,	Table 1551, p. 1252
Analog Reserved 2 Register (n=0–6)	Port0: 0x000A011C, Port1: 0x000A091C, Port2: 0x000A111C, Port3: 0x000A191C, Port4: 0x000A211C, Port5: 0x000A291C,	Table 1552, p. 1252
PCIe/USB3 Control Register (n=0–6)	Port0: 0x000A0120, Port1: 0x000A0920, Port2: 0x000A1120, Port3: 0x000A1920, Port4: 0x000A2120, Port5: 0x000A2920,	Table 1553, p. 1252
Lane Alignment Control Register (n=0–6)	Port0: 0x000A0124, Port1: 0x000A0924, Port2: 0x000A1124, Port3: 0x000A1924, Port4: 0x000A2124, Port5: 0x000A2924,	Table 1554, p. 1252
Analog Test Point Selection Register (n=0–6)	Port0: 0x000A0128, Port1: 0x000A0928, Port2: 0x000A1128, Port3: 0x000A1928, Port4: 0x000A2128, Port5: 0x000A2928,	Table 1555, p. 1253
150M Clock Control 0 Register (n=0–6)	Port0: 0x000A012C, Port1: 0x000A092C, Port2: 0x000A112C, Port3: 0x000A192C, Port4: 0x000A212C, Port5: 0x000A292C,	Table 1556, p. 1253
150M Clock Control 1 Register (n=0–6)	Port0: 0x000A0130, Port1: 0x000A0930, Port2: 0x000A1130, Port3: 0x000A1930, Port4: 0x000A2130, Port5: 0x000A2930,	Table 1557, p. 1253
Regulator Output Control 0 Register (n=0–6)	Port0: 0x000A0134, Port1: 0x000A0934, Port2: 0x000A1134, Port3: 0x000A1934, Port4: 0x000A2134, Port5: 0x000A2934,	Table 1558, p. 1253
Regulator Output Control 1 Register (n=0–6)	Port0: 0x000A0138, Port1: 0x000A0938, Port2: 0x000A1138, Port3: 0x000A1938, Port4: 0x000A2138, Port5: 0x000A2938,	Table 1559, p. 1254

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Miscellaneous Control 0 Register (n=0–6)	Port0: 0x000A013C, Port1: 0x000A093C, Port2: 0x000A113C, Port3: 0x000A193C, Port4: 0x000A213C, Port5: 0x000A293C,	Table 1560, p. 1254
RX Control 1 Register (n=0–6)	Port0: 0x000A0140, Port1: 0x000A0940, Port2: 0x000A1140, Port3: 0x000A1940, Port4: 0x000A2140, Port5: 0x000A2940,	Table 1561, p. 1256
Interface and Digital Reference Clock Control Register (n=0–6)	Port0: 0x000A0144, Port1: 0x000A0944, Port2: 0x000A1144, Port3: 0x000A1944, Port4: 0x000A2144, Port5: 0x000A2944,	Table 1562, p. 1256
Power Control Register (n=0–6)	Port0: 0x000A0148, Port1: 0x000A0948, Port2: 0x000A1148, Port3: 0x000A1948, Port4: 0x000A2148, Port5: 0x000A2948,	Table 1563, p. 1257
Feedback Divider Register (n=0–6)	Port0: 0x000A014C, Port1: 0x000A094C, Port2: 0x000A114C, Port3: 0x000A194C, Port4: 0x000A214C, Port5: 0x000A294C,	Table 1564, p. 1257
Forward Divider Register (n=0–6)	Port0: 0x000A0150, Port1: 0x000A0950, Port2: 0x000A1150, Port3: 0x000A1950, Port4: 0x000A2150, Port5: 0x000A2950,	Table 1565, p. 1257
Transmitter/Receiver Speed Divider Force Register (n=0–6)	Port0: 0x000A0154, Port1: 0x000A0954, Port2: 0x000A1154, Port3: 0x000A1954, Port4: 0x000A2154, Port5: 0x000A2954,	Table 1566, p. 1257
PLL Bandwidth Adjust Register (n=0–6)	Port0: 0x000A0158, Port1: 0x000A0958, Port2: 0x000A1158, Port3: 0x000A1958, Port4: 0x000A2158, Port5: 0x000A2958,	Table 1567, p. 1258
Calibration Enable Control Register (n=0–6)	Port0: 0x000A015C, Port1: 0x000A095C, Port2: 0x000A115C, Port3: 0x000A195C, Port4: 0x000A215C, Port5: 0x000A295C,	Table 1568, p. 1258

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
TX DCC Control Register (n=0–6)	Port0: 0x000A0160, Port1: 0x000A0960, Port2: 0x000A1160, Port3: 0x000A1960, Port4: 0x000A2160, Port5: 0x000A2960,	Table 1569, p. 1259
RX DCC Edge and Center Control Register (n=0–6)	Port0: 0x000A0164, Port1: 0x000A0964, Port2: 0x000A1164, Port3: 0x000A1964, Port4: 0x000A2164, Port5: 0x000A2964,	Table 1570, p. 1259
RX CLK Align 90 and TX Idle Calibration Control Register (n=0–6)	Port0: 0x000A0168, Port1: 0x000A0968, Port2: 0x000A1168, Port3: 0x000A1968, Port4: 0x000A2168, Port5: 0x000A2968,	Table 1571, p. 1259
Sampler and Process Calibration Control Register (n=0–6)	Port0: 0x000A016C, Port1: 0x000A096C, Port2: 0x000A116C, Port3: 0x000A196C, Port4: 0x000A216C, Port5: 0x000A296C,	Table 1572, p. 1260
Process Threshold Setting Register (n=0–6)	Port0: 0x000A0170, Port1: 0x000A0970, Port2: 0x000A1170, Port3: 0x000A1970, Port4: 0x000A2170, Port5: 0x000A2970,	Table 1573, p. 1260
Transmitter Slew Rate Control Register (n=0–6)	Port0: 0x000A0174, Port1: 0x000A0974, Port2: 0x000A1174, Port3: 0x000A1974, Port4: 0x000A2174, Port5: 0x000A2974,	Table 1574, p. 1260
Transmitter Amplitude and Emphasis Control Register (n=0–6)	Port0: 0x000A0178, Port1: 0x000A0978, Port2: 0x000A1178, Port3: 0x000A1978, Port4: 0x000A2178, Port5: 0x000A2978,	Table 1575, p. 1261
SSC Control Register (n=0–6)	Port0: 0x000A017C, Port1: 0x000A097C, Port2: 0x000A117C, Port3: 0x000A197C, Port4: 0x000A217C, Port5: 0x000A297C,	Table 1576, p. 1261
SSC Step Register (n=0–6)	Port0: 0x000A0180, Port1: 0x000A0980, Port2: 0x000A1180, Port3: 0x000A1980, Port4: 0x000A2180, Port5: 0x000A2980,	Table 1577, p. 1261

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
DTL Control Register (n=0–6)	Port0: 0x000A0184, Port1: 0x000A0984, Port2: 0x000A1184, Port3: 0x000A1984, Port4: 0x000A2184, Port5: 0x000A2984,	Table 1578, p. 1262
Offset Phase Control Register (n=0–6)	Port0: 0x000A0188, Port1: 0x000A0988, Port2: 0x000A1188, Port3: 0x000A1988, Port4: 0x000A2188, Port5: 0x000A2988,	Table 1579, p. 1262
Receiver Frequency Offset Extraction Register (n=0–6)	Port0: 0x000A018C, Port1: 0x000A098C, Port2: 0x000A118C, Port3: 0x000A198C, Port4: 0x000A218C, Port5: 0x000A298C,	Table 1580, p. 1262
Receiver Frequency Offset Read Register (n=0–6)	Port0: 0x000A0190, Port1: 0x000A0990, Port2: 0x000A1190, Port3: 0x000A1990, Port4: 0x000A2190, Port5: 0x000A2990,	Table 1581, p. 1262
Receiver Offset Control Register (n=0–6)	Port0: 0x000A0194, Port1: 0x000A0994, Port2: 0x000A1194, Port3: 0x000A1994, Port4: 0x000A2194, Port5: 0x000A2994,	Table 1582, p. 1263
Receiver Clamping Control Register (n=0–6)	Port0: 0x000A0198, Port1: 0x000A0998, Port2: 0x000A1198, Port3: 0x000A1998, Port4: 0x000A2198, Port5: 0x000A2998,	Table 1583, p. 1263
DFE Miscellaneous Control Register (n=0–6)	Port0: 0x000A019C, Port1: 0x000A099C, Port2: 0x000A119C, Port3: 0x000A199C, Port4: 0x000A219C, Port5: 0x000A299C,	Table 1584, p. 1263
Data Phase Offset Control Register (n=0–6)	Port0: 0x000A01A0, Port1: 0x000A09A0, Port2: 0x000A11A0, Port3: 0x000A19A0, Port4: 0x000A21A0, Port5: 0x000A29A0,	Table 1585, p. 1264
Adapted FFE Capacitor/Counter Control Register (n=0–6)	Port0: 0x000A01A4, Port1: 0x000A09A4, Port2: 0x000A11A4, Port3: 0x000A19A4, Port4: 0x000A21A4, Port5: 0x000A29A4,	Table 1586, p. 1264

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Error Counter Constraint Control Register (n=0–6)	Port0: 0x000A01A8, Port1: 0x000A09A8, Port2: 0x000A11A8, Port3: 0x000A19A8, Port4: 0x000A21A8, Port5: 0x000A29A8,	Table 1587, p. 1264
DFE Timer Register (n=0–6)	Port0: 0x000A01AC, Port1: 0x000A09AC, Port2: 0x000A11AC, Port3: 0x000A19AC, Port4: 0x000A21AC, Port5: 0x000A29AC,	Table 1588, p. 1265
Eye Threshold Mode Selection Register (n=0–6)	Port0: 0x000A01B0, Port1: 0x000A09B0, Port2: 0x000A11B0, Port3: 0x000A19B0, Port4: 0x000A21B0, Port5: 0x000A29B0,	Table 1589, p. 1266
RX SSC Extraction Register (n=0–6)	Port0: 0x000A01B4, Port1: 0x000A09B4, Port2: 0x000A11B4, Port3: 0x000A19B4, Port4: 0x000A21B4, Port5: 0x000A29B4,	Table 1590, p. 1266
Digital Test Bus Configuration Register (n=0–6)	Port0: 0x000A01B8, Port1: 0x000A09B8, Port2: 0x000A11B8, Port3: 0x000A19B8, Port4: 0x000A21B8, Port5: 0x000A29B8,	Table 1591, p. 1266
External Squelch Calibration Control Register (n=0–6)	Port0: 0x000A01BC, Port1: 0x000A09BC, Port2: 0x000A11BC, Port3: 0x000A19BC, Port4: 0x000A21BC, Port5: 0x000A29BC,	Table 1592, p. 1267
PLL Calibration Control 0 Register (n=0–6)	Port0: 0x000A01C0, Port1: 0x000A09C0, Port2: 0x000A11C0, Port3: 0x000A19C0, Port4: 0x000A21C0, Port5: 0x000A29C0,	Table 1593, p. 1267
PLL Calibration Control 1 Register (n=0–6)	Port0: 0x000A01C4, Port1: 0x000A09C4, Port2: 0x000A11C4, Port3: 0x000A19C4, Port4: 0x000A21C4, Port5: 0x000A29C4,	Table 1594, p. 1267
Squelch Glitch Filter Control Register (n=0–6)	Port0: 0x000A01C8, Port1: 0x000A09C8, Port2: 0x000A11C8, Port3: 0x000A19C8, Port4: 0x000A21C8, Port5: 0x000A29C8,	Table 1595, p. 1268

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Miscellaneous Control 1 Register (n=0–6)	Port0: 0x000A01CC, Port1: 0x000A09CC, Port2: 0x000A11CC, Port3: 0x000A19CC, Port4: 0x000A21CC, Port5: 0x000A29CC,	Table 1596, p. 1268
Miscellaneous Control 1 Register (n=0–6)	Port0: 0x000A01D0, Port1: 0x000A09D0, Port2: 0x000A11D0, Port3: 0x000A19D0, Port4: 0x000A21D0, Port5: 0x000A29D0,	Table 1597, p. 1269
Squelch Calibration Register (n=0–6)	Port0: 0x000A01D4, Port1: 0x000A09D4, Port2: 0x000A11D4, Port3: 0x000A19D4, Port4: 0x000A21D4, Port5: 0x000A29D4,	Table 1598, p. 1269
Reg0p9 Control Register (n=0–6)	Port0: 0x000A01D8, Port1: 0x000A09D8, Port2: 0x000A11D8, Port3: 0x000A19D8, Port4: 0x000A21D8, Port5: 0x000A29D8,	Table 1599, p. 1270
Clk150M Control 2 Register (n=0–6)	Port0: 0x000A01DC, Port1: 0x000A09DC, Port2: 0x000A11DC, Port3: 0x000A19DC, Port4: 0x000A21DC, Port5: 0x000A29DC,	Table 1600, p. 1270
Cal_reg19 Register (n=0–6)	Port0: 0x000A01E0, Port1: 0x000A09E0, Port2: 0x000A11E0, Port3: 0x000A19E0, Port4: 0x000A21E0, Port5: 0x000A29E0,	Table 1601, p. 1271
Impedance Calibration Result Register (n=0–6)	Port0: 0x000A01E4, Port1: 0x000A09E4, Port2: 0x000A11E4, Port3: 0x000A19E4, Port4: 0x000A21E4, Port5: 0x000A29E4,	Table 1602, p. 1271
Impedance Calibration Result Register (n=0–6)	Port0: 0x000A01E8, Port1: 0x000A09E8, Port2: 0x000A11E8, Port3: 0x000A19E8, Port4: 0x000A21E8, Port5: 0x000A29E8,	Table 1603, p. 1271
F1 Calibration Result Register (n=0–6)	Port0: 0x000A01EC, Port1: 0x000A09EC, Port2: 0x000A11EC, Port3: 0x000A19EC, Port4: 0x000A21EC, Port5: 0x000A29EC,	Table 1604, p. 1271

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
RX DCC Calibration Binary Register (n=0–6)	Port0: 0x000A01F0, Port1: 0x000A09F0, Port2: 0x000A11F0, Port3: 0x000A19F0, Port4: 0x000A21F0, Port5: 0x000A29F0,	Table 1605, p. 1272
TX DCC Calibration Binary Register (n=0–6)	Port0: 0x000A01F4, Port1: 0x000A09F4, Port2: 0x000A11F4, Port3: 0x000A19F4, Port4: 0x000A21F4, Port5: 0x000A29F4,	Table 1606, p. 1272
Squelch Control 2 Register (n=0–6)	Port0: 0x000A01F8, Port1: 0x000A09F8, Port2: 0x000A11F8, Port3: 0x000A19F8, Port4: 0x000A21F8, Port5: 0x000A29F8,	Table 1607, p. 1272
RX Control 7 Register (n=0–6)	Port0: 0x000A0200, Port1: 0x000A0A00, Port2: 0x000A1200, Port3: 0x000A1A00, Port4: 0x000A2200, Port5: 0x000A2A00,	Table 1608, p. 1272
AC JTAG Control 0 Register (n=0–6)	Port0: 0x000A0204, Port1: 0x000A0A04, Port2: 0x000A1204, Port3: 0x000A1A04, Port4: 0x000A2204, Port5: 0x000A2A04,	Table 1609, p. 1273
DFE F1p5 and F2p5 Control and Other Register (n=0–6)	Port0: 0x000A0208, Port1: 0x000A0A08, Port2: 0x000A1208, Port3: 0x000A1A08, Port4: 0x000A2208, Port5: 0x000A2A08,	Table 1610, p. 1273
FFE Control Register (n=0–6)	Port0: 0x000A020C, Port1: 0x000A0A0C, Port2: 0x000A120C, Port3: 0x000A1A0C, Port4: 0x000A220C, Port5: 0x000A2A0C,	Table 1611, p. 1273
DFE 14 Register (n=0–6)	Port0: 0x000A0210, Port1: 0x000A0A10, Port2: 0x000A1210, Port3: 0x000A1A10, Port4: 0x000A2210, Port5: 0x000A2A10,	Table 1612, p. 1274
Frame Detection Control 0 Register (n=0–6)	Port0: 0x000A0214, Port1: 0x000A0A14, Port2: 0x000A1214, Port3: 0x000A1A14, Port4: 0x000A2214, Port5: 0x000A2A14,	Table 1613, p. 1274

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
PLL Calibration Control 11 Register (n=0–6)	Port0: 0x000A0218, Port1: 0x000A0A18, Port2: 0x000A1218, Port3: 0x000A1A18, Port4: 0x000A2218, Port5: 0x000A2A18,	Table 1614, p. 1274
Frame Detection Control 2 Register (n=0–6)	Port0: 0x000A021C, Port1: 0x000A0A1C, Port2: 0x000A121C, Port3: 0x000A1A1C, Port4: 0x000A221C, Port5: 0x000A2A1C,	Table 1615, p. 1275
Frame Detection Control 3 Register (n=0–6)	Port0: 0x000A0220, Port1: 0x000A0A20, Port2: 0x000A1220, Port3: 0x000A1A20, Port4: 0x000A2220, Port5: 0x000A2A20,	Table 1616, p. 1275
TX Training Control 0 Register (n=0–6)	Port0: 0x000A022C, Port1: 0x000A0A2C, Port2: 0x000A122C, Port3: 0x000A1A2C, Port4: 0x000A222C, Port5: 0x000A2A2C,	Table 1617, p. 1275
TRX Training Control 1 Register (n=0–6)	Port0: 0x000A0230, Port1: 0x000A0A30, Port2: 0x000A1230, Port3: 0x000A1A30, Port4: 0x000A2230, Port5: 0x000A2A30,	Table 1618, p. 1276
RX Training Control 0 Register (n=0–6)	Port0: 0x000A0234, Port1: 0x000A0A34, Port2: 0x000A1234, Port3: 0x000A1A34, Port4: 0x000A2234, Port5: 0x000A2A34,	Table 1619, p. 1277
RX Training Control 1 Register (n=0–6)	Port0: 0x000A0238, Port1: 0x000A0A38, Port2: 0x000A1238, Port3: 0x000A1A38, Port4: 0x000A2238, Port5: 0x000A2A38,	Table 1620, p. 1277
RX Training Control 2 Register (n=0–6)	Port0: 0x000A023C, Port1: 0x000A0A3C, Port2: 0x000A123C, Port3: 0x000A1A3C, Port4: 0x000A223C, Port5: 0x000A2A3C,	Table 1621, p. 1277
RX Training Control 3 Register (n=0–6)	Port0: 0x000A0240, Port1: 0x000A0A40, Port2: 0x000A1240, Port3: 0x000A1A40, Port4: 0x000A2240, Port5: 0x000A2A40,	Table 1622, p. 1277

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
RX Training Control 4 Register (n=0–6)	Port0: 0x000A0244, Port1: 0x000A0A44, Port2: 0x000A1244, Port3: 0x000A1A44, Port4: 0x000A2244, Port5: 0x000A2A44,	Table 1623, p. 1278
RX Training Control 5 Register (n=0–6)	Port0: 0x000A0248, Port1: 0x000A0A48, Port2: 0x000A1248, Port3: 0x000A1A48, Port4: 0x000A2248, Port5: 0x000A2A48,	Table 1624, p. 1278
RX Training Control 6 Register (n=0–6)	Port0: 0x000A024C, Port1: 0x000A0A4C, Port2: 0x000A124C, Port3: 0x000A1A4C, Port4: 0x000A224C, Port5: 0x000A2A4C,	Table 1625, p. 1279
RX Training Control 7 Register (n=0–6)	Port0: 0x000A0250, Port1: 0x000A0A50, Port2: 0x000A1250, Port3: 0x000A1A50, Port4: 0x000A2250, Port5: 0x000A2A50,	Table 1626, p. 1279
RX Training Control 8 Register (n=0–6)	Port0: 0x000A0254, Port1: 0x000A0A54, Port2: 0x000A1254, Port3: 0x000A1A54, Port4: 0x000A2254, Port5: 0x000A2A54,	Table 1627, p. 1279
RX Training Control 9 Register (n=0–6)	Port0: 0x000A0258, Port1: 0x000A0A58, Port2: 0x000A1258, Port3: 0x000A1A58, Port4: 0x000A2258, Port5: 0x000A2A58,	Table 1628, p. 1280
RX Training Control 10 Register (n=0–6)	Port0: 0x000A025C, Port1: 0x000A0A5C, Port2: 0x000A125C, Port3: 0x000A1A5C, Port4: 0x000A225C, Port5: 0x000A2A5C,	Table 1629, p. 1280
RX Training Control 11 Register (n=0–6)	Port0: 0x000A0260, Port1: 0x000A0A60, Port2: 0x000A1260, Port3: 0x000A1A60, Port4: 0x000A2260, Port5: 0x000A2A60,	Table 1630, p. 1280
RX Training Control 12 Register (n=0–6)	Port0: 0x000A0264, Port1: 0x000A0A64, Port2: 0x000A1264, Port3: 0x000A1A64, Port4: 0x000A2264, Port5: 0x000A2A64,	Table 1631, p. 1281

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
TX Training Control 0 Register (n=0–6)	Port0: 0x000A0268, Port1: 0x000A0A68, Port2: 0x000A1268, Port3: 0x000A1A68, Port4: 0x000A2268, Port5: 0x000A2A68,	Table 1632, p. 1281
TX Training Control 1 Register (n=0–6)	Port0: 0x000A026C, Port1: 0x000A0A6C, Port2: 0x000A126C, Port3: 0x000A1A6C, Port4: 0x000A226C, Port5: 0x000A2A6C,	Table 1633, p. 1281
TRX Training Control 2 Register (n=0–6)	Port0: 0x000A0270, Port1: 0x000A0A70, Port2: 0x000A1270, Port3: 0x000A1A70, Port4: 0x000A2270, Port5: 0x000A2A70,	Table 1634, p. 1282
TRX Training Control 4 Register (n=0–6)	Port0: 0x000A0278, Port1: 0x000A0A78, Port2: 0x000A1278, Port3: 0x000A1A78, Port4: 0x000A2278, Port5: 0x000A2A78,	Table 1635, p. 1282
DFE Adaptation Control 0 Register (n=0–6)	Port0: 0x000A027C, Port1: 0x000A0A7C, Port2: 0x000A127C, Port3: 0x000A1A7C, Port4: 0x000A227C, Port5: 0x000A2A7C,	Table 1636, p. 1282
DFE Adaptation Control 1 Register (n=0–6)	Port0: 0x000A0280, Port1: 0x000A0A80, Port2: 0x000A1280, Port3: 0x000A1A80, Port4: 0x000A2280, Port5: 0x000A2A80,	Table 1637, p. 1283
DFE Adaptation Control 2 Register (n=0–6)	Port0: 0x000A0284, Port1: 0x000A0A84, Port2: 0x000A1284, Port3: 0x000A1A84, Port4: 0x000A2284, Port5: 0x000A2A84,	Table 1638, p. 1283
TX Amplitude Setting for PIPE Register (n=0–6)	Port0: 0x000A0288, Port1: 0x000A0A88, Port2: 0x000A1288, Port3: 0x000A1A88, Port4: 0x000A2288, Port5: 0x000A2A88,	Table 1639, p. 1283
TX Margin Selection Register (n=0–6)	Port0: 0x000A0290, Port1: 0x000A0A90, Port2: 0x000A1290, Port3: 0x000A1A90, Port4: 0x000A2290, Port5: 0x000A2A90,	Table 1640, p. 1283

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Tx_Amp_Adj and Tx_Emph_En Register (n=0–6)	Port0: 0x000A0294, Port1: 0x000A0A94, Port2: 0x000A1294, Port3: 0x000A1A94, Port4: 0x000A2294, Port5: 0x000A2A94,	Table 1641, p. 1284
TX Amplitude Index Limit Register (n=0–6)	Port0: 0x000A029C, Port1: 0x000A0A9C, Port2: 0x000A129C, Port3: 0x000A1A9C, Port4: 0x000A229C, Port5: 0x000A2A9C,	Table 1642, p. 1284
TX Emphasis Index Minimum Value Register (n=0–6)	Port0: 0x000A02A0, Port1: 0x000A0AA0, Port2: 0x000A12A0, Port3: 0x000A1AA0, Port4: 0x000A22A0, Port5: 0x000A2AA0,	Table 1643, p. 1285
TRX Training Control 5 Register (n=0–6)	Port0: 0x000A02A4, Port1: 0x000A0AA4, Port2: 0x000A12A4, Port3: 0x000A1AA4, Port4: 0x000A22A4, Port5: 0x000A2AA4,	Table 1644, p. 1285
Interrupt 1 Register (n=0–6)	Port0: 0x000A02AC, Port1: 0x000A0AAC, Port2: 0x000A12AC, Port3: 0x000A1AAC, Port4: 0x000A22AC, Port5: 0x000A2AAC,	Table 1645, p. 1285
Interrupt 3 Register (n=0–6)	Port0: 0x000A02B4, Port1: 0x000A0AB4, Port2: 0x000A12B4, Port3: 0x000A1AB4, Port4: 0x000A22B4, Port5: 0x000A2AB4,	Table 1646, p. 1286
Interrupt 5 Register (n=0–6)	Port0: 0x000A02BC, Port1: 0x000A0ABC, Port2: 0x000A12BC, Port3: 0x000A1ABC, Port4: 0x000A22BC, Port5: 0x000A2ABC,	Table 1647, p. 1287
TX Amplitude Default Register (n=0–6)	Port0: 0x000A02DC, Port1: 0x000A0ADC, Port2: 0x000A12DC, Port3: 0x000A1ADC, Port4: 0x000A22DC, Port5: 0x000A2ADC,	Table 1648, p. 1287
Value of Signals to Analog Selection Register (n=0–6)	Port0: 0x000A02E0, Port1: 0x000A0AE0, Port2: 0x000A12E0, Port3: 0x000A1AE0, Port4: 0x000A22E0, Port5: 0x000A2AE0,	Table 1649, p. 1287

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
EOM Phase Control Register (n=0–6)	Port0: 0x000A02EC, Port1: 0x000A0AEC, Port2: 0x000A12EC, Port3: 0x000A1AEC, Port4: 0x000A22EC, Port5: 0x000A2AEC,	Table 1650, p. 1288
EOM Calibration Control 0 Register (n=0–6)	Port0: 0x000A0308, Port1: 0x000A0B08, Port2: 0x000A1308, Port3: 0x000A1B08, Port4: 0x000A2308, Port5: 0x000A2B08,	Table 1651, p. 1288
EOM Calibration Control 1 Register (n=0–6)	Port0: 0x000A030C, Port1: 0x000A0B0C, Port2: 0x000A130C, Port3: 0x000A1B0C, Port4: 0x000A230C, Port5: 0x000A2B0C,	Table 1652, p. 1288
TX Train Result Register (n=0–6)	Port0: 0x000A0310, Port1: 0x000A0B10, Port2: 0x000A1310, Port3: 0x000A1B10, Port4: 0x000A2310, Port5: 0x000A2B10,	Table 1653, p. 1288
TX Train Result Register (n=0–6)	Port0: 0x000A0314, Port1: 0x000A0B14, Port2: 0x000A1314, Port3: 0x000A1B14, Port4: 0x000A2314, Port5: 0x000A2B14,	Table 1654, p. 1289
Adapted Edge Result Register (n=0–6)	Port0: 0x000A0318, Port1: 0x000A0B18, Port2: 0x000A1318, Port3: 0x000A1B18, Port4: 0x000A2318, Port5: 0x000A2B18,	Table 1655, p. 1289
TX Train Preset Register (n=0–6)	Port0: 0x000A031C, Port1: 0x000A0B1C, Port2: 0x000A131C, Port3: 0x000A1B1C, Port4: 0x000A231C, Port5: 0x000A2B1C,	Table 1656, p. 1290
Phase Offset Range Register (n=0–6)	Port0: 0x000A0320, Port1: 0x000A0B20, Port2: 0x000A1320, Port3: 0x000A1B20, Port4: 0x000A2320, Port5: 0x000A2B20,	Table 1657, p. 1290
Saved DFE Values 0 Register (n=0–6)	Port0: 0x000A0324, Port1: 0x000A0B24, Port2: 0x000A1324, Port3: 0x000A1B24, Port4: 0x000A2324, Port5: 0x000A2B24,	Table 1658, p. 1290

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Saved DFE Values 1 Register (n=0–6)	Port0: 0x000A0328, Port1: 0x000A0B28, Port2: 0x000A1328, Port3: 0x000A1B28, Port4: 0x000A2328, Port5: 0x000A2B28,	Table 1659, p. 1291
Saved DFE Values 2 Register (n=0–6)	Port0: 0x000A032C, Port1: 0x000A0B2C, Port2: 0x000A132C, Port3: 0x000A1B2C, Port4: 0x000A232C, Port5: 0x000A2B2C,	Table 1660, p. 1291
RX Train Default Values Register (n=0–6)	Port0: 0x000A0330, Port1: 0x000A0B30, Port2: 0x000A1330, Port3: 0x000A1B30, Port4: 0x000A2330, Port5: 0x000A2B30,	Table 1661, p. 1291
Status Detection Timeout Control Register (n=0–6)	Port0: 0x000A0334, Port1: 0x000A0B34, Port2: 0x000A1334, Port3: 0x000A1B34, Port4: 0x000A2334, Port5: 0x000A2B34,	Table 1662, p. 1291
TX Amplitude Default Register (n=0–6)	Port0: 0x000A0338, Port1: 0x000A0B38, Port2: 0x000A1338, Port3: 0x000A1B38, Port4: 0x000A2338, Port5: 0x000A2B38,	Table 1663, p. 1292
TX Amplitude Default Register (n=0–6)	Port0: 0x000A033C, Port1: 0x000A0B3C, Port2: 0x000A133C, Port3: 0x000A1B3C, Port4: 0x000A233C, Port5: 0x000A2B3C,	Table 1664, p. 1292
DFE Adaptation Control 3 Register (n=0–6)	Port0: 0x000A0340, Port1: 0x000A0B40, Port2: 0x000A1340, Port3: 0x000A1B40, Port4: 0x000A2340, Port5: 0x000A2B40,	Table 1665, p. 1292
RX Training Control 14 Register (n=0–6)	Port0: 0x000A0348, Port1: 0x000A0B48, Port2: 0x000A1348, Port3: 0x000A1B48, Port4: 0x000A2348, Port5: 0x000A2B48,	Table 1666, p. 1293
RX Training Control 15 Register (n=0–6)	Port0: 0x000A034C, Port1: 0x000A0B4C, Port2: 0x000A134C, Port3: 0x000A1B4C, Port4: 0x000A234C, Port5: 0x000A2B4C,	Table 1667, p. 1293

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
RX Training Control 16 Register (n=0–6)	Port0: 0x000A0350, Port1: 0x000A0B50, Port2: 0x000A1350, Port3: 0x000A1B50, Port4: 0x000A2350, Port5: 0x000A2B50,	Table 1668, p. 1293
TX Training Control 6 Register (n=0–6)	Port0: 0x000A035C, Port1: 0x000A0B5C, Port2: 0x000A135C, Port3: 0x000A1B5C, Port4: 0x000A235C, Port5: 0x000A2B5C,	Table 1669, p. 1293
TX Training Control 7 Register (n=0–6)	Port0: 0x000A0360, Port1: 0x000A0B60, Port2: 0x000A1360, Port3: 0x000A1B60, Port4: 0x000A2360, Port5: 0x000A2B60,	Table 1670, p. 1293
TX Training Control 7 Register (n=0–6)	Port0: 0x000A0364, Port1: 0x000A0B64, Port2: 0x000A1364, Port3: 0x000A1B64, Port4: 0x000A2364, Port5: 0x000A2B64,	Table 1671, p. 1294
TRX Training Control 6 Register (n=0–6)	Port0: 0x000A036C, Port1: 0x000A0B6C, Port2: 0x000A136C, Port3: 0x000A1B6C, Port4: 0x000A236C, Port5: 0x000A2B6C,	Table 1672, p. 1294
Calibration Control 16 Register (n=0–6)	Port0: 0x000A0370, Port1: 0x000A0B70, Port2: 0x000A1370, Port3: 0x000A1B70, Port4: 0x000A2370, Port5: 0x000A2B70,	Table 1673, p. 1294
VDD Calibration Control 4 Register (n=0–6)	Port0: 0x000A0374, Port1: 0x000A0B74, Port2: 0x000A1374, Port3: 0x000A1B74, Port4: 0x000A2374, Port5: 0x000A2B74,	Table 1674, p. 1295
F1 Buffer Offset Calibration Control 0 Register (n=0–6)	Port0: 0x000A0378, Port1: 0x000A0B78, Port2: 0x000A1378, Port3: 0x000A1B78, Port4: 0x000A2378, Port5: 0x000A2B78,	Table 1675, p. 1295
TX Default 4 Register (n=0–6)	Port0: 0x000A037C, Port1: 0x000A0B7C, Port2: 0x000A137C, Port3: 0x000A1B7C, Port4: 0x000A237C, Port5: 0x000A2B7C,	Table 1676, p. 1295

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
PLL Calibration Control 2 Register (n=0–6)	Port0: 0x000A0380, Port1: 0x000A0B80, Port2: 0x000A1380, Port3: 0x000A1B80, Port4: 0x000A2380, Port5: 0x000A2B80,	Table 1677, p. 1295
PLL Calibration Control 3 Register (n=0–6)	Port0: 0x000A0384, Port1: 0x000A0B84, Port2: 0x000A1384, Port3: 0x000A1B84, Port4: 0x000A2384, Port5: 0x000A2B84,	Table 1678, p. 1296
PLL Calibration Control 4 Register (n=0–6)	Port0: 0x000A0388, Port1: 0x000A0B88, Port2: 0x000A1388, Port3: 0x000A1B88, Port4: 0x000A2388, Port5: 0x000A2B88,	Table 1679, p. 1296
PLL Calibration Control 5 Register (n=0–6)	Port0: 0x000A038C, Port1: 0x000A0B8C, Port2: 0x000A138C, Port3: 0x000A1B8C, Port4: 0x000A238C, Port5: 0x000A2B8C,	Table 1680, p. 1296
PLL Calibration Control 6 Register (n=0–6)	Port0: 0x000A0390, Port1: 0x000A0B90, Port2: 0x000A1390, Port3: 0x000A1B90, Port4: 0x000A2390, Port5: 0x000A2B90,	Table 1681, p. 1296
PLL Calibration Control 8 Register (n=0–6)	Port0: 0x000A0398, Port1: 0x000A0B98, Port2: 0x000A1398, Port3: 0x000A1B98, Port4: 0x000A2398, Port5: 0x000A2B98,	Table 1682, p. 1297
FFE Control 2 Register (n=0–6)	Port0: 0x000A039C, Port1: 0x000A0B9C, Port2: 0x000A139C, Port3: 0x000A1B9C, Port4: 0x000A239C, Port5: 0x000A2B9C,	Table 1683, p. 1297
Miscellaneous Control 3 Register (n=0–6)	Port0: 0x000A03A0, Port1: 0x000A0BA0, Port2: 0x000A13A0, Port3: 0x000A1BA0, Port4: 0x000A23A0, Port5: 0x000A2BA0,	Table 1684, p. 1297
Miscellaneous Control 4 Register (n=0–6)	Port0: 0x000A03A4, Port1: 0x000A0BA4, Port2: 0x000A13A4, Port3: 0x000A1BA4, Port4: 0x000A23A4, Port5: 0x000A2BA4,	Table 1685, p. 1298

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
TRX Training Control 7 Register (n=0–6)	Port0: 0x000A03A8, Port1: 0x000A0BA8, Port2: 0x000A13A8, Port3: 0x000A1BA8, Port4: 0x000A23A8, Port5: 0x000A2BA8,	Table 1686, p. 1298
TRX Training Control 8 Register (n=0–6)	Port0: 0x000A03AC, Port1: 0x000A0BAC, Port2: 0x000A13AC, Port3: 0x000A1BAC, Port4: 0x000A23AC, Port5: 0x000A2BAC,	Table 1687, p. 1298
DFE Step Size 2 Register (n=0–6)	Port0: 0x000A03B0, Port1: 0x000A0BB0, Port2: 0x000A13B0, Port3: 0x000A1BB0, Port4: 0x000A23B0, Port5: 0x000A2BB0,	Table 1688, p. 1298
TRX Training Control 10 Register (n=0–6)	Port0: 0x000A03B4, Port1: 0x000A0BB4, Port2: 0x000A13B4, Port3: 0x000A1BB4, Port4: 0x000A23B4, Port5: 0x000A2BB4,	Table 1689, p. 1300
PLL Calibration Control Register (n=0–6)	Port0: 0x000A03B8, Port1: 0x000A0BB8, Port2: 0x000A13B8, Port3: 0x000A1BB8, Port4: 0x000A23B8, Port5: 0x000A2BB8,	Table 1690, p. 1300
Align 90 Calibration Control 0 Register (n=0–6)	Port0: 0x000A03BC, Port1: 0x000A0BBC, Port2: 0x000A13BC, Port3: 0x000A1BBC, Port4: 0x000A23BC, Port5: 0x000A2BBC,	Table 1691, p. 1301
Digital Test Bus Configuration Register (n=0–6)	Port0: 0x000A03C0, Port1: 0x000A0BC0, Port2: 0x000A13C0, Port3: 0x000A1BC0, Port4: 0x000A23C0, Port5: 0x000A2BC0,	Table 1692, p. 1301
Digital Test Bus Register (n=0–6)	Port0: 0x000A03C4, Port1: 0x000A0BC4, Port2: 0x000A13C4, Port3: 0x000A1BC4, Port4: 0x000A23C4, Port5: 0x000A2BC4,	Table 1693, p. 1301
Analog Control 1 Register (n=0–6)	Port0: 0x000A03C8, Port1: 0x000A0BC8, Port2: 0x000A13C8, Port3: 0x000A1BC8, Port4: 0x000A23C8, Port5: 0x000A2BC8,	Table 1694, p. 1301

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Analog Control 2 Register (n=0–6)	Port0: 0x000A03CC, Port1: 0x000A0BCC, Port2: 0x000A13CC, Port3: 0x000A1BCC, Port4: 0x000A23CC, Port5: 0x000A2BCC,	Table 1695, p. 1302
Analog Control 3 Register (n=0–6)	Port0: 0x000A03D0, Port1: 0x000A0BD0, Port2: 0x000A13D0, Port3: 0x000A1BD0, Port4: 0x000A23D0, Port5: 0x000A2BD0,	Table 1696, p. 1302
Analog Control 4 Register (n=0–6)	Port0: 0x000A03D4, Port1: 0x000A0BD4, Port2: 0x000A13D4, Port3: 0x000A1BD4, Port4: 0x000A23D4, Port5: 0x000A2BD4,	Table 1697, p. 1302
TX Training Control 25 Register (n=0–6)	Port0: 0x000A03D8, Port1: 0x000A0BD8, Port2: 0x000A13D8, Port3: 0x000A1BD8, Port4: 0x000A23D8, Port5: 0x000A2BD8,	Table 1698, p. 1303
TX Default 5 Register (n=0–6)	Port0: 0x000A03DC, Port1: 0x000A0BDC, Port2: 0x000A13DC, Port3: 0x000A1BDC, Port4: 0x000A23DC, Port5: 0x000A2BDC,	Table 1699, p. 1303
Clk150m Control 3 Register (n=0–6)	Port0: 0x000A03E0, Port1: 0x000A0BE0, Port2: 0x000A13E0, Port3: 0x000A1BE0, Port4: 0x000A23E0, Port5: 0x000A2BE0,	Table 1700, p. 1303
Signal Detection Calibration Register (n=0–6)	Port0: 0x000A03E4, Port1: 0x000A0BE4, Port2: 0x000A13E4, Port3: 0x000A1BE4, Port4: 0x000A23E4, Port5: 0x000A2BE4,	Table 1701, p. 1303
Analog Control 5 Register (n=0–6)	Port0: 0x000A03E8, Port1: 0x000A0BE8, Port2: 0x000A13E8, Port3: 0x000A1BE8, Port4: 0x000A23E8, Port5: 0x000A2BE8,	Table 1702, p. 1304
TX Default 7 Register (n=0–6)	Port0: 0x000A03EC, Port1: 0x000A0BEC, Port2: 0x000A13EC, Port3: 0x000A1BEC, Port4: 0x000A23EC, Port5: 0x000A2BEC,	Table 1703, p. 1304

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
TRX Training Control 9 Register (n=0–6)	Port0: 0x000A03F0, Port1: 0x000A0BF0, Port2: 0x000A13F0, Port3: 0x000A1BF0, Port4: 0x000A23F0, Port5: 0x000A2BF0,	Table 1704, p. 1304
TX Default 6 Register (n=0–6)	Port0: 0x000A03F4, Port1: 0x000A0BF4, Port2: 0x000A13F4, Port3: 0x000A1BF4, Port4: 0x000A23F4, Port5: 0x000A2BF4,	Table 1705, p. 1305
Squelch OOB Register (n=0–6)	Port0: 0x000A03F8, Port1: 0x000A0BF8, Port2: 0x000A13F8, Port3: 0x000A1BF8, Port4: 0x000A23F8, Port5: 0x000A2BF8,	Table 1706, p. 1305
Squelch Detect Calibration and VDD Calibration Result Register (n=0–6)	Port0: 0x000A0400, Port1: 0x000A0C00, Port2: 0x000A1400, Port3: 0x000A1C00, Port4: 0x000A2400, Port5: 0x000A2C00,	Table 1707, p. 1305
Align 90 Calibration Result Register (n=0–6)	Port0: 0x000A0404, Port1: 0x000A0C04, Port2: 0x000A1404, Port3: 0x000A1C04, Port4: 0x000A2404, Port5: 0x000A2C04,	Table 1708, p. 1306
DCC Calibration Result Register (n=0–6)	Port0: 0x000A0408, Port1: 0x000A0C08, Port2: 0x000A1408, Port3: 0x000A1C08, Port4: 0x000A2408, Port5: 0x000A2C08,	Table 1709, p. 1306
Calibration Result Register (n=0–6)	Port0: 0x000A040C, Port1: 0x000A0C0C, Port2: 0x000A140C, Port3: 0x000A1C0C, Port4: 0x000A240C, Port5: 0x000A2C0C,	Table 1710, p. 1306
RX DCC Calibration Result Register (n=0–6)	Port0: 0x000A0410, Port1: 0x000A0C10, Port2: 0x000A1410, Port3: 0x000A1C10, Port4: 0x000A2410, Port5: 0x000A2C10,	Table 1711, p. 1306
Sampler Calibration Result Register (n=0–6)	Port0: 0x000A0414, Port1: 0x000A0C14, Port2: 0x000A1414, Port3: 0x000A1C14, Port4: 0x000A2414, Port5: 0x000A2C14,	Table 1712, p. 1307

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
CDR Control Register (n=0–6)	Port0: 0x000A0418, Port1: 0x000A0C18, Port2: 0x000A1418, Port3: 0x000A1C18, Port4: 0x000A2418, Port5: 0x000A2C18,	Table 1713, p. 1307
CDR Control Register (n=0–6)	Port0: 0x000A041C, Port1: 0x000A0C1C, Port2: 0x000A141C, Port3: 0x000A1C1C, Port4: 0x000A241C, Port5: 0x000A2C1C,	Table 1714, p. 1308
CDR Control Register (n=0–6)	Port0: 0x000A0420, Port1: 0x000A0C20, Port2: 0x000A1420, Port3: 0x000A1C20, Port4: 0x000A2420, Port5: 0x000A2C20,	Table 1715, p. 1308
CDR Control Register (n=0–6)	Port0: 0x000A0424, Port1: 0x000A0C24, Port2: 0x000A1424, Port3: 0x000A1C24, Port4: 0x000A2424, Port5: 0x000A2C24,	Table 1716, p. 1309
RX Train Midpoint Register (n=0–6)	Port0: 0x000A0428, Port1: 0x000A0C28, Port2: 0x000A1428, Port3: 0x000A1C28, Port4: 0x000A2428, Port5: 0x000A2C28,	Table 1717, p. 1309
RX Train Midpoint Register (n=0–6)	Port0: 0x000A042C, Port1: 0x000A0C2C, Port2: 0x000A142C, Port3: 0x000A1C2C, Port4: 0x000A242C, Port5: 0x000A2C2C,	Table 1718, p. 1309
TX Training Control 9 Register (n=0–6)	Port0: 0x000A0430, Port1: 0x000A0C30, Port2: 0x000A1430, Port3: 0x000A1C30, Port4: 0x000A2430, Port5: 0x000A2C30,	Table 1719, p. 1309
TX Training Control 10 Register (n=0–6)	Port0: 0x000A0434, Port1: 0x000A0C34, Port2: 0x000A1434, Port3: 0x000A1C34, Port4: 0x000A2434, Port5: 0x000A2C34,	Table 1720, p. 1310
TX Training Control 11 Register (n=0–6)	Port0: 0x000A0438, Port1: 0x000A0C38, Port2: 0x000A1438, Port3: 0x000A1C38, Port4: 0x000A2438, Port5: 0x000A2C38,	Table 1721, p. 1310

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
TX Training Control 12 Register (n=0–6)	Port0: 0x000A043C, Port1: 0x000A0C3C, Port2: 0x000A143C, Port3: 0x000A1C3C, Port4: 0x000A243C, Port5: 0x000A2C3C,	Table 1722, p. 1310
G1 Setting 3 Register (n=0–6)	Port0: 0x000A0440, Port1: 0x000A0C40, Port2: 0x000A1440, Port3: 0x000A1C40, Port4: 0x000A2440, Port5: 0x000A2C40,	Table 1723, p. 1310
G1 Setting 4 Register (n=0–6)	Port0: 0x000A0444, Port1: 0x000A0C44, Port2: 0x000A1444, Port3: 0x000A1C44, Port4: 0x000A2444, Port5: 0x000A2C44,	Table 1724, p. 1311
G2 Setting 3 Register (n=0–6)	Port0: 0x000A0448, Port1: 0x000A0C48, Port2: 0x000A1448, Port3: 0x000A1C48, Port4: 0x000A2448, Port5: 0x000A2C48,	Table 1725, p. 1312
G2 Setting 4 Register (n=0–6)	Port0: 0x000A044C, Port1: 0x000A0C4C, Port2: 0x000A144C, Port3: 0x000A1C4C, Port4: 0x000A244C, Port5: 0x000A2C4C,	Table 1726, p. 1312
G3 Setting 3 Register (n=0–6)	Port0: 0x000A0450, Port1: 0x000A0C50, Port2: 0x000A1450, Port3: 0x000A1C50, Port4: 0x000A2450, Port5: 0x000A2C50,	Table 1727, p. 1313
G3 Setting 4 Register (n=0–6)	Port0: 0x000A0454, Port1: 0x000A0C54, Port2: 0x000A1454, Port3: 0x000A1C54, Port4: 0x000A2454, Port5: 0x000A2C54,	Table 1728, p. 1314
G4_setting_3 Register (n=0–6)	Port0: 0x000A0458, Port1: 0x000A0C58, Port2: 0x000A1458, Port3: 0x000A1C58, Port4: 0x000A2458, Port5: 0x000A2C58,	Table 1729, p. 1314
G4 Setting 4 Register (n=0–6)	Port0: 0x000A045C, Port1: 0x000A0C5C, Port2: 0x000A145C, Port3: 0x000A1C5C, Port4: 0x000A245C, Port5: 0x000A2C5C,	Table 1730, p. 1315

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
PCIe TX Preset Table 0 Register (n=0–6)	Port0: 0x000A0460, Port1: 0x000A0C60, Port2: 0x000A1460, Port3: 0x000A1C60, Port4: 0x000A2460, Port5: 0x000A2C60,	Table 1731, p. 1315
PCIe TX Preset Table 1 Register (n=0–6)	Port0: 0x000A0464, Port1: 0x000A0C64, Port2: 0x000A1464, Port3: 0x000A1C64, Port4: 0x000A2464, Port5: 0x000A2C64,	Table 1732, p. 1316
PCIe TX Preset Index Register (n=0–6)	Port0: 0x000A0468, Port1: 0x000A0C68, Port2: 0x000A1468, Port3: 0x000A1C68, Port4: 0x000A2468, Port5: 0x000A2C68,	Table 1733, p. 1316
TX Power Protection Control Register (n=0–6)	Port0: 0x000A046C, Port1: 0x000A0C6C, Port2: 0x000A146C, Port3: 0x000A1C6C, Port4: 0x000A246C, Port5: 0x000A2C6C,	Table 1734, p. 1316
DFE Control 27 Register (n=0–6)	Port0: 0x000A0470, Port1: 0x000A0C70, Port2: 0x000A1470, Port3: 0x000A1C70, Port4: 0x000A2470, Port5: 0x000A2C70,	Table 1735, p. 1317
RX Training for Over Boost Register (n=0–6)	Port0: 0x000A0474, Port1: 0x000A0C74, Port2: 0x000A1474, Port3: 0x000A1C74, Port4: 0x000A2474, Port5: 0x000A2C74,	Table 1736, p. 1317
TX Training for Over Boost Register (n=0–6)	Port0: 0x000A0478, Port1: 0x000A0C78, Port2: 0x000A1478, Port3: 0x000A1C78, Port4: 0x000A2478, Port5: 0x000A2C78,	Table 1737, p. 1318
RX Training Mode 0 Register (n=0–6)	Port0: 0x000A047C, Port1: 0x000A0C7C, Port2: 0x000A147C, Port3: 0x000A1C7C, Port4: 0x000A247C, Port5: 0x000A2C7C,	Table 1738, p. 1318
PCIe RX Preset Hint 1 Register (n=0–6)	Port0: 0x000A0480, Port1: 0x000A0C80, Port2: 0x000A1480, Port3: 0x000A1C80, Port4: 0x000A2480, Port5: 0x000A2C80,	Table 1739, p. 1318

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
PCIe RX Preset Hint 2 Register (n=0–6)	Port0: 0x000A0484, Port1: 0x000A0C84, Port2: 0x000A1484, Port3: 0x000A1C84, Port4: 0x000A2484, Port5: 0x000A2C84,	Table 1740, p. 1318
PCIe RX Preset Hint 3 Register (n=0–6)	Port0: 0x000A0488, Port1: 0x000A0C88, Port2: 0x000A1488, Port3: 0x000A1C88, Port4: 0x000A2488, Port5: 0x000A2C88,	Table 1741, p. 1319
PCIe RX Preset Hint 4 Register (n=0–6)	Port0: 0x000A048C, Port1: 0x000A0C8C, Port2: 0x000A148C, Port3: 0x000A1C8C, Port4: 0x000A248C, Port5: 0x000A2C8C,	Table 1742, p. 1319
PCIe RX Preset Hint 5 Register (n=0–6)	Port0: 0x000A0490, Port1: 0x000A0C90, Port2: 0x000A1490, Port3: 0x000A1C90, Port4: 0x000A2490, Port5: 0x000A2C90,	Table 1743, p. 1320
PCIe RX Preset Hint 6 Register (n=0–6)	Port0: 0x000A0494, Port1: 0x000A0C94, Port2: 0x000A1494, Port3: 0x000A1C94, Port4: 0x000A2494, Port5: 0x000A2C94,	Table 1744, p. 1320
PCIe RX Preset Hint 7 Register (n=0–6)	Port0: 0x000A0498, Port1: 0x000A0C98, Port2: 0x000A1498, Port3: 0x000A1C98, Port4: 0x000A2498, Port5: 0x000A2C98,	Table 1745, p. 1320
DFE Control 28 Register (n=0–6)	Port0: 0x000A049C, Port1: 0x000A0C9C, Port2: 0x000A149C, Port3: 0x000A1C9C, Port4: 0x000A249C, Port5: 0x000A2C9C,	Table 1746, p. 1320
F1_BUF_OS Adjust 0 Register (n=0–6)	Port0: 0x000A04A0, Port1: 0x000A0CA0, Port2: 0x000A14A0, Port3: 0x000A1CA0, Port4: 0x000A24A0, Port5: 0x000A2CA0,	Table 1747, p. 1321
F1_BUF_OS Adjust 1 Register (n=0–6)	Port0: 0x000A04A4, Port1: 0x000A0CA4, Port2: 0x000A14A4, Port3: 0x000A1CA4, Port4: 0x000A24A4, Port5: 0x000A2CA4,	Table 1748, p. 1321

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
F1_BUF_OS Adjust 2 Register (n=0–6)	Port0: 0x000A04A8, Port1: 0x000A0CA8, Port2: 0x000A14A8, Port3: 0x000A1CA8, Port4: 0x000A24A8, Port5: 0x000A2CA8,	Table 1749, p. 1322
F1_BUF_OS Adjust 3 Register (n=0–6)	Port0: 0x000A04AC, Port1: 0x000A0CAC, Port2: 0x000A14AC, Port3: 0x000A1CAC, Port4: 0x000A24AC, Port5: 0x000A2CAC,	Table 1750, p. 1322
TX Default 8 Register (n=0–6)	Port0: 0x000A04B0, Port1: 0x000A0CB0, Port2: 0x000A14B0, Port3: 0x000A1CB0, Port4: 0x000A24B0, Port5: 0x000A2CB0,	Table 1751, p. 1322
TX Default 9 Register (n=0–6)	Port0: 0x000A04B4, Port1: 0x000A0CB4, Port2: 0x000A14B4, Port3: 0x000A1CB4, Port4: 0x000A24B4, Port5: 0x000A2CB4,	Table 1752, p. 1323
TX Default 10 Register (n=0–6)	Port0: 0x000A04B8, Port1: 0x000A0CB8, Port2: 0x000A14B8, Port3: 0x000A1CB8, Port4: 0x000A24B8, Port5: 0x000A2CB8,	Table 1753, p. 1323
TX Default 11 Register (n=0–6)	Port0: 0x000A04BC, Port1: 0x000A0CBC, Port2: 0x000A14BC, Port3: 0x000A1CBC, Port4: 0x000A24BC, Port5: 0x000A2CBC,	Table 1754, p. 1323
DFE 30 Register (n=0–6)	Port0: 0x000A04C0, Port1: 0x000A0CC0, Port2: 0x000A14C0, Port3: 0x000A1CC0, Port4: 0x000A24C0, Port5: 0x000A2CC0,	Table 1755, p. 1323
Internal Digital Reserved 0 Register (n=0–6)	Port0: 0x000A05EC, Port1: 0x000A0DEC, Port2: 0x000A15EC, Port3: 0x000A1DEC, Port4: 0x000A25EC, Port5: 0x000A2DEC,	Table 1756, p. 1324
Analog Reserved 0 Register (n=0–6)	Port0: 0x000A05F0, Port1: 0x000A0DF0, Port2: 0x000A15F0, Port3: 0x000A1DF0, Port4: 0x000A25F0, Port5: 0x000A2DF0,	Table 1757, p. 1324

Document Classification: Proprietary Information

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Internal CID 0 Register (n=0–6)	Port0: 0x000A05F4, Port1: 0x000A0DF4, Port2: 0x000A15F4, Port3: 0x000A1DF4, Port4: 0x000A25F4, Port5: 0x000A2DF4,	Table 1758, p. 1324
Internal CID 1 Register (n=0–6)	Port0: 0x000A05F8, Port1: 0x000A0DF8, Port2: 0x000A15F8, Port3: 0x000A1DF8, Port4: 0x000A25F8, Port5: 0x000A2DF8,	Table 1759, p. 1324
Lane Configuration 0 Register (n=0–1)	Port0: 0x000A0600, Port1: 0x000A0E00	Table 1760, p. 1324
Lane Configuration 1 Register (n=0–1)	Port0: 0x000A0604, Port1: 0x000A0E04	Table 1761, p. 1326
Lane Status 0 Register (n=0–1)	Port0: 0x000A0608, Port1: 0x000A0E08	Table 1762, p. 1327
Lane Status 1 Register (n=0–1)	Port0: 0x000A060C, Port1: 0x000A0E0C	Table 1763, p. 1329
Lane Status 2 Register (n=0–1)	Port0: 0x000A0610, Port1: 0x000A0E10	Table 1764, p. 1330
Lane Beacon Control Register (n=0–1)	Port0: 0x000A0614, Port1: 0x000A0E14	Table 1765, p. 1331
Lane Configuration 2 Register (n=0–1)	Port0: 0x000A0618, Port1: 0x000A0E18	Table 1766, p. 1332
Lane Configuration 3 Register (n=0–1)	Port0: 0x000A061C, Port1: 0x000A0E1C	Table 1767, p. 1333
Lane Configuration 4 Register (n=0–1)	Port0: 0x000A0620, Port1: 0x000A0E20	Table 1768, p. 1333
Lane Configuration 5 Register (n=0–1)	Port0: 0x000A0624, Port1: 0x000A0E24	Table 1769, p. 1335
Lane Status 3 Register (n=0–1)	Port0: 0x000A0628, Port1: 0x000A0E28	Table 1770, p. 1335
Lane TX Common Mode Configuration Register (n=0–1)	Port0: 0x000A062C, Port1: 0x000A0E2C	Table 1771, p. 1336
USB Datapath Configuration 1 Register (n=0–1)	Port0: 0x000A0684, Port1: 0x000A0E84	Table 1772, p. 1336
Lane Equalization Configuration 0 Register (n=0–1)	Port0: 0x000A069C, Port1: 0x000A0E9C	Table 1773, p. 1337
Lane Equalization Configuration 1 Register (n=0–1)	Port0: 0x000A06A0, Port1: 0x000A0EA0	Table 1774, p. 1337
Lane Equalization Configuration 2 Register (n=0–1)	Port0: 0x000A06A4, Port1: 0x000A0EA4	Table 1775, p. 1337
Lane Equalization Preset Configuration 0 Register (n=0–1)	Port0: 0x000A06A8, Port1: 0x000A0EA8	Table 1776, p. 1337
Lane Equalization Preset Configuration 1 Register (n=0–1)	Port0: 0x000A06AC, Port1: 0x000A0EAC	Table 1777, p. 1338

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Lane Equalization Preset Configuration 2 Register (n=0–1)	Port0: 0x000A06B0, Port1: 0x000A0EB0	Table 1778, p. 1338
Lane Equalization Preset Configuration 3 Register (n=0–1)	Port0: 0x000A06B4, Port1: 0x000A0EB4	Table 1779, p. 1338
Lane Equalization Preset Configuration 4 Register (n=0–1)	Port0: 0x000A06B8, Port1: 0x000A0EB8	Table 1780, p. 1339
Lane Equalization Preset Configuration 5 Register (n=0–1)	Port0: 0x000A06BC, Port1: 0x000A0EBC	Table 1781, p. 1339
Lane Equalization Preset Configuration 6 Register (n=0–1)	Port0: 0x000A06C0, Port1: 0x000A0EC0	Table 1782, p. 1339
Lane Equalization Preset Configuration 7 Register (n=0–1)	Port0: 0x000A06C4, Port1: 0x000A0EC4	Table 1783, p. 1339
Lane Equalization Preset Configuration 8 Register (n=0–1)	Port0: 0x000A06C8, Port1: 0x000A0EC8	Table 1784, p. 1340
Lane Equalization Preset Configuration 9 Register (n=0–1)	Port0: 0x000A06CC, Port1: 0x000A0ECC	Table 1785, p. 1340
Lane Equalization Preset Configuration 10 Register (n=0–1)	Port0: 0x000A06D0, Port1: 0x000A0ED0	Table 1786, p. 1340
Lane Equalization Preset Configuration 11 Register (n=0–1)	Port0: 0x000A06D4, Port1: 0x000A0ED4	Table 1787, p. 1340
Lane Equalization Preset Configuration 12 Register (n=0–1)	Port0: 0x000A06D8, Port1: 0x000A0ED8	Table 1788, p. 1341
Lane Equalization Preset Configuration 13 Register (n=0–1)	Port0: 0x000A06DC, Port1: 0x000A0EDC	Table 1789, p. 1341
Lane Equalization Preset Configuration 14 Register (n=0–1)	Port0: 0x000A06E0, Port1: 0x000A0EE0	Table 1790, p. 1341
Lane Equalization Preset Configuration 15 Register (n=0–1)	Port0: 0x000A06E4, Port1: 0x000A0EE4	Table 1791, p. 1341
Lane Equalization Preset Configuration 16 Register (n=0–1)	Port0: 0x000A06E8, Port1: 0x000A0EE8	Table 1792, p. 1342
Lane Equalization Preset Configuration 17 Register (n=0–1)	Port0: 0x000A06EC, Port1: 0x000A0EEC	Table 1793, p. 1342
Lane Equalization Coefficient Max Setting 0 Register (n=0–1)	Port0: 0x000A06F0, Port1: 0x000A0EF0	Table 1794, p. 1342
Lane Equalization Coefficient Max Setting 1 Register (n=0–1)	Port0: 0x000A06F4, Port1: 0x000A0EF4	Table 1795, p. 1342
Lane Equalization Remote Setting Register (n=0–1)	Port0: 0x000A06F8, Port1: 0x000A0EF8	Table 1796, p. 1343
Lane Equalization Status Register (n=0–1)	Port0: 0x000A06FC, Port1: 0x000A0EFC	Table 1797, p. 1343
PIPE Revision Register (n=0–1)	Port0: 0x000A0700, Port1: 0x000A0F00	Table 1798, p. 1343
Reset and Clock Control Register (n=0–1)	Port0: 0x000A0704, Port1: 0x000A0F04	Table 1799, p. 1343
Test Mode Control Register (n=0–1)	Port0: 0x000A0708, Port1: 0x000A0F08	Table 1800, p. 1344
Clock Source Low Register (n=0–1)	Port0: 0x000A070C, Port1: 0x000A0F0C	Table 1801, p. 1346

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
Clock Source High Register (n=0–1)	Port0: 0x000A0710, Port1: 0x000A0F10	Table 1802, p. 1347
Trigger Register (n=0–1)	Port0: 0x000A0714, Port1: 0x000A0F14	Table 1803, p. 1348
Miscellaneous Control Register (n=0–1)	Port0: 0x000A0718, Port1: 0x000A0F18	Table 1804, p. 1349
CLKREQ# Timing Control Register (n=0–1)	Port0: 0x000A071C, Port1: 0x000A0F1C	Table 1805, p. 1351
Datapath Configuration Register (n=0–1)	Port0: 0x000A0720, Port1: 0x000A0F20	Table 1806, p. 1352
Symbol Alignment Good Symbol Weight Register (n=0–1)	Port0: 0x000A0724, Port1: 0x000A0F24	Table 1807, p. 1353
Symbol Alignment Aligned COM Weight Register (n=0–1)	Port0: 0x000A0728, Port1: 0x000A0F28	Table 1808, p. 1353
Symbol Alignment Bad Symbol Weight Register (n=0–1)	Port0: 0x000A072C, Port1: 0x000A0F2C	Table 1809, p. 1353
Symbol Alignment Misaligned COM Weight Register (n=0–1)	Port0: 0x000A0730, Port1: 0x000A0F30	Table 1810, p. 1354
Symbol Alignment Squelch Detect Register (n=0–1)	Port0: 0x000A0734, Port1: 0x000A0F34	Table 1811, p. 1354
Symbol Alignment Comma Pattern Register (n=0–1)	Port0: 0x000A0738, Port1: 0x000A0F38	Table 1812, p. 1354
Power Management Timing Parameter 1 Register (n=0–1)	Port0: 0x000A0740, Port1: 0x000A0F40	Table 1813, p. 1355
Power Management Timing Parameter 2 Register (n=0–1)	Port0: 0x000A0744, Port1: 0x000A0F44	Table 1814, p. 1355
Counter Lane and Type Register (n=0–1)	Port0: 0x000A0748, Port1: 0x000A0F48	Table 1815, p. 1355
Counter Low Register (n=0–1)	Port0: 0x000A074C, Port1: 0x000A0F4C	Table 1816, p. 1356
Counter High Register (n=0–1)	Port0: 0x000A0750, Port1: 0x000A0F50	Table 1817, p. 1356
PM Serdes Control Override Register (n=0–1)	Port0: 0x000A0754, Port1: 0x000A0F54	Table 1818, p. 1357
PM Datapath and Clock Control Override Register (n=0–1)	Port0: 0x000A0758, Port1: 0x000A0F58	Table 1819, p. 1357
Low Frequency Configuration Register (n=0–1)	Port0: 0x000A075C, Port1: 0x000A0F5C	Table 1820, p. 1357
Block Alignment Good Block Weight Register (n=0–1)	Port0: 0x000A0760, Port1: 0x000A0F60	Table 1821, p. 1357
Block Alignment Aligned EIEOS Weight Register (n=0–1)	Port0: 0x000A0764, Port1: 0x000A0F64	Table 1822, p. 1358
Block Alignment Bad Block Weight Register (n=0–1)	Port0: 0x000A0768, Port1: 0x000A0F68	Table 1823, p. 1358
Block Alignment Misaligned EIEOS Weight Register (n=0–1)	Port0: 0x000A076C, Port1: 0x000A0F6C	Table 1824, p. 1358
Block Alignment Squelch Detect Register (n=0–1)	Port0: 0x000A0770, Port1: 0x000A0F70	Table 1825, p. 1358

Copyright © 2015 Marvell

Table 1480: Summary Map Table for the Common PHY and PIPE Interface Registers (Continued)

Register Name	Offset	Table and Page
BIST Control Input Register (n=0–1)	Port0: 0x000A0780, Port1: 0x000A0F80	Table 1826, p. 1359
BIST Configuration Input Register (n=0–1)	Port0: 0x000A0784, Port1: 0x000A0F84	Table 1827, p. 1359
BIST Data Input Register (n=0–1)	Port0: 0x000A0788, Port1: 0x000A0F88	Table 1828, p. 1360
BIST LANE and TYPE Register (n=0–1)	Port0: 0x000A078C, Port1: 0x000A0F8C	Table 1829, p. 1360
BIST Loggin Window Start Cycle Register (n=0–1)	Port0: 0x000A0790, Port1: 0x000A0F90	Table 1830, p. 1360
BIST Loggin Window Length Register (n=0–1)	Port0: 0x000A0794, Port1: 0x000A0F94	Table 1831, p. 1360
BIST Result Mask Low Register (n=0–1)	Port0: 0x000A0798, Port1: 0x000A0F98	Table 1832, p. 1361
BIST Result Mask High Register (n=0–1)	Port0: 0x000A079C, Port1: 0x000A0F9C	Table 1833, p. 1361
BIST Result Low Register (n=0–1)	Port0: 0x000A07A0, Port1: 0x000A0FA0	Table 1834, p. 1361
BIST Result High Register (n=0–1)	Port0: 0x000A07A4, Port1: 0x000A0FA4	Table 1835, p. 1361
BIST Sequencer Configuration Register (n=0–1)	Port0: 0x000A07A8, Port1: 0x000A0FA8	Table 1836, p. 1361
BIST LFSR Seed Register (n=0–1)	Port0: 0x000A07AC, Port1: 0x000A0FAC	Table 1837, p. 1362

A.24.1 Common PHY and PIPE Interface

Table 1481: Chip ID Register (n=0–6)
Offset: Port0: 0x000A0000, Port1: 0x000A0800, Port2: 0x000A1000, Port3: 0x000A1800, Port4: 0x000A2000, Port5: 0x000A2800,
Offset Formula: 0x000A0000+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	CID0[7:4]	RO 0x0	Phy technology. Defines the process node used for the PHY design. {CID0 comes from the IP.}
11:8	CID0[3:0]	RO 0x0	PHY type.Defines the type of PHY Defines the type of PHY {CID0 comes from the IP.}
7:4	CID1[7:4]	RO 0x0	Major Revision. Defines the all layer change for a given PHY type. {CID1 comes from the SOC.}

Copyright © 2015 Marvell

Table 1481: Chip ID Register (n=0–6) (Continued)
Offset: Port0: 0x000A0000, Port1: 0x000A0800, Port2: 0x000A1000, Port3: 0x000A1800, Port4: 0x000A2000, Port5: 0x000A2800,
Offset Formula: 0x000A0000+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	CID1[3:0]	RO 0x0	Minor revision. Defines a metal revision to a given phy type. This field is incremented every time a metal revision is taped out for the PHY for a given major revision. {CID1 comes from the SOC.}

Table 1482: Power and PLL Control Register (n=0–6)
Offset: Port0: 0x000A0004, Port1: 0x000A0804, Port2: 0x000A1004, Port3: 0x000A1804, Port4: 0x000A2004, Port5: 0x000A2804,
Offset Formula: 0x000A0004+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x1	Reserved
14	PU_PLL	RW 0x1	Power Up PLL. This bit only works in the PHY Isolate Mode. For more information, refer to 1.7, PHY Isolate Mode. 0 = Power down 1 = Power up
13	PU_RX	RW 0x1	Power Up Receiver. This bit only works in the PHY Isolate Mode. For more information, refer to 1.7, PHY Isolate Mode. 0 = Power down 1 = Power up
12	PU_TX	RW 0x1	Power Up Transmitter. This bit only works in the PHY Isolate Mode. For more information, refer to 1.7, PHY Isolate Mode. 0 = Power down 1 = Power up
11	Reserved	RSVD 0x1	Reserved
10	PU_DFE	RW 0x1	Power Up Decision Feedback Equalization (DFE). 0 = Power down 1 = Power up
9	Reserved	RSVD 0x0	Reserved
8	PLL_LOCK	RO 0x0	PLL Lock Indicator This field is not needed for regular operation. The internal FSMs use this flag to assert PHY ready signal. 0 = PLL is not in lock. Real time signal. 1 = PLL is in lock.

Copyright © 2015 Marvell

Table 1482: Power and PLL Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A0004, Port1: 0x000A0804, Port2: 0x000A1004, Port3: 0x000A1804, Port4: 0x000A2004, Port5: 0x000A2804,

Offset Formula: 0x000A0004+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
7:5	PHY_MODE[2:0]	RW 0x3	PHY Mode. These bits control the mode selection Note: The mode not mentioned here are not supported. 0 = SATA 1 = SAS 2 = FC 3 = PCIE 4 = SERDES 5 = USB3

Copyright © 2015 Marvell

Table 1482: Power and PLL Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A0004, Port1: 0x000A0804, Port2: 0x000A1004, Port3: 0x000A1804, Port4: 0x000A2004, Port5: 0x000A2804,
Offset Formula: 0x000A0004+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
4:0	REF_FREF_SEL[4:0]	RW 0x0	Reference Frequency Select. These bits indicate the reference clock speed. Unit is MHz. For SATA/SAS: 0x00 = 20 0x01 = 25 0x02 = 30 0x03 = 40 0x04 = 50 0x05 = 62.5 0x06 = 75 0x07 = 100 0x08 = 125 0x09 = 150 0x0A = 156.25 0x0B = 250 0x0C = 31.25 0x0D = 31 0x0E = 33.33 For FC: 0x00 = 25 0x01 = 50 0x02 = 53.125 0x03 = 62.5 0x04 = 75 0x05 = 100 0x06 = 106.25 0x07 = 125 0x08 = 150 0x09 = 156.25 0x0A = 212.5 0x0B = 250 0x0C = 31.25 For PCIe: 0x00 = 100MHz 0x01 = 20 0x02 = 25 0x03 = 40 0x04 = 50 0x05 = 62.5 0x06 = 75 0x07 = 125 0x08 = 150 0x09 = 156.25 0x0A = 250 0x0B = 30 0x0C = 31.25 0x0D = 26 0x0E = 19.2 0x0F = 38.4

Copyright © 2015 Marvell

Table 1482: Power and PLL Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A0004, Port1: 0x000A0804, Port2: 0x000A1004, Port3: 0x000A1804, Port4: 0x000A2004, Port5: 0x000A2804,

Offset Formula: 0x000A0004+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
4:0	REF_FREF_SEL[4:0]	RW 0x0	For USB 3.0: 0x00 = 100 0x01 = 20 0x02 = 25 0x03 = 40 0x04 = 50 0x05 = 62.5 0x06 = 75 0x07 = 125 0x08 = 150 0x09 = 156.25 0x0A = 250 0x0B = 30 0x0C = 31.25 0x0D = 26 0x0E = 19.2 0x0F = 38.4 For SERDES: 0x00 = 20 0x01 = 25 0x02 = 30 0x03 = 40 0x04 = 50 0x05 = 53.125 0x06 = 62.5 0x07 = 75 0x08 = 100 0x09 = 106.25 0x0A = 125 0x0B = 150 0x0C = 156.25 0x0D = 212.5 0x0E = 250

Table 1483: KVCO Calibration Control Register (n=0–6)

Offset: Port0: 0x000A0008, Port1: 0x000A0808, Port2: 0x000A1008, Port3: 0x000A1808, Port4: 0x000A2008, Port5: 0x000A2808,

Offset Formula: 0x000A0008+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	CAL_START	RW 0x0	Calibration Start. A rising edge (0->1 transition) on this signal starts the main calibration state machine.

Document Classification: Proprietary Information

Table 1483: KVCO Calibration Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A0008, Port1: 0x000A0808, Port2: 0x000A1008, Port3: 0x000A1808, Port4: 0x000A2008, Port5: 0x000A2808,
Offset Formula: 0x000A0008+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
14	CAL_DONE	RO 0x0	Calibration Done. This is the main auto calibration state machine done signal. The firmware should wait for this bit to set to 1 before moving on to a normal operation. 0 = Calibration sequence is in progress. 1 = Calibration sequence is done.
13	Reserved	RSVD 0x0	Reserved
12	USE_MAX_PLL_RATE	RW 0x0	Use Maximum PLL Rate. This signal indicates whether to use speed configuration 1 or speed configuration 2. With speed configuration 1, lowering the bit rates may lower the PLL VCO speed to conserve power. When using speed configuration 2, the PLL VCO runs at the same speed for all the standard bit rates. 0 = Speed configuration 1: lower power for the lower bit rates but takes longer to change speeds 1 = Speed configuration 2: more power at the lower bit rates but can switch bit rates more quickly
11:10	Reserved	RSVD 0x0	Reserved
9	EXT_FORCE_CAL_DONE	RW 0x0	Force Calibration Done. This signal forces the main calibration done signal to be asserted and is used only for testing. This signal can be used to assert done if the calibration state machine does not complete. (This control has no effect on the individual calibration circuits.)
8	SPEED_EXT_EN	RW 0x0	Speed_pll External Value Enable. This field is used to select the value from the register or the auto-calibration result 0 = Use the auto-calibration value. 1 = Use SPEED value from register
7:2	SPEED[5:0]	RW 0xC	Speed_pll external overwrite value.
1	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1484: Impedance Calibration Control Register (n=0–6)

Offset: Port0: 0x000A000C, Port1: 0x000A080C, Port2: 0x000A100C, Port3: 0x000A180C, Port4: 0x000A200C, Port5: 0x000A280C,

Offset Formula: 0x000A000C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	EXT_TXIMP_EN	RW 0x0	External Impedance Enable for TX This field is used to select the impedance value from the register or the auto-calibration result 0 = Use value from the auto-calibration result 1 = Use value from the registers EXT_TXIMP[4:0]
14:10	EXT_TXIMP[4:0]	RW 0x10	External Transmitter impedance. When EXT_TXIMP_EN = 1, this value is used in place of the auto-calibration impedance result for the transmitter impedance control circuits.
9:5	TXIMP[4:0]	RO 0x0	TX impedance auto-calibration result
4	EXT_RXIMP_EN	RW 0x0	External Impedance Enable for RX This field is used to select the impedance value from the register or the auto-calibration result 0 = Use value from the auto-calibration result 1 = Use value from the registers EXT_RXIMP[3:0]
3:2	Reserved	RSVD 0x0	Reserved
1	SSC_DSPREAD_TX	RW 0x1	TX Spread Spectrum Clock Down-spread. Down-spread SSC Selection. 0 = Center-spread 1 = Down-spread
0	SSC_DSPREAD_RX	RW 0x1	Spread Spectrum Clock Down-spread for RX frequency offset extraction

Table 1485: Impedance and TX SSC Control Register (n=0–6)

Offset: Port0: 0x000A0010, Port1: 0x000A0810, Port2: 0x000A1010, Port3: 0x000A1810, Port4: 0x000A2010, Port5: 0x000A2810,

Offset Formula: 0x000A0010+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:11	EXT_RXIMP[4:0]	RW 0x10	External Receiver impedance. When EXT_RXIMP_EN = 1, this value is used in place of the auto-calibration impedance result for the receiver impedance control circuits.
10:6	RXIMP[4:0]	RO 0x0	RX impedance auto-calibration result
5:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1486: Transmitter Frequency Offset Register (n=0–6)

Offset: Port0: 0x000A0014, Port1: 0x000A0814, Port2: 0x000A1014, Port3: 0x000A1814, Port4: 0x000A2014, Port5: 0x000A2814,

Offset Formula: 0x000A0014+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1487: Squelch and FFE Setting Register (n=0–6)

Offset: Port0: 0x000A0018, Port1: 0x000A0818, Port2: 0x000A1018, Port3: 0x000A1818, Port4: 0x000A2018, Port5: 0x000A2818,

Offset Formula: 0x000A0018+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14	SQ_DETECTED	RO 0x0	Squelch Detected 0 = Squelch is not detected by analog circuit 1 = Squelch is detected by analog circuit
13:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:8	SQ_THRESH_IN[3:0]	RW 0x5	Squelch Detector Threshold Setting 0 = V50mv: 50mv 1 = V75mv: 75mv 2 = V100mv: 100mv 3 = V125mv: 125mv 4 = V150mv: 150mv 5 = V175mv: 175mv 6 = V200mv: 200mv 7 = V225mv: 225mv 8 = V250mv: 250mv 9 = V275mv: 275mv 10 = V300mv: 300mv 11 = V325mv: 325mv 12 = V350mv: 350mv 13 = V375mv: 375mv 14 = V400mv: 400mv 15 = V425mv: 425mv
7:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1488: DFE Resolution Control Register (n=0–6)

Offset: Port0: 0x000A001C, Port1: 0x000A081C, Port2: 0x000A101C, Port3: 0x000A181C, Port4: 0x000A201C, Port5: 0x000A281C,

Offset Formula: 0x000A001C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	DFE_RES_FORCE	RW 0x0	Force DFE resolution
14	DFE_SQ_EN	RW 0x1	Decision Feedback Equalizer (DFE) Squelch Enable. The DFE allows the Squelch detector to freeze DFE. 0 = DFE continues updating even if Squelch circuit indicates the loss of signal 1 = DFE updating is frozen when the Squelch circuit indicates the loss of signal
13	DFE_PAT_DIS	RW 0x1	Decision Feedback Equalization (DFE) Pattern Disable. 0 = Enable (DFE freezes the adaptation when a pattern is not valid for the DFE training) 1 = Disable
12	DFE_LOCK	RO 0x0	Decision Feedback Equalization (DFE) Lock. The DFE lock is detected by checking that the DFE taps do not change significantly (less that 2 LSB variation on all taps) within DFE_LOCK_LEN[1:0]. The link layer logic should not use this signal as a part of its state machines. It may be possible for the DFE lock signal to occasionally report unlocked even though the DFE is locked. If any of the DFE taps change more that 2 LSBs, then the DFE lock signal is de-asserted. While this is not expected to occur, it is possible that the combination of training variation, pattern variation, temperature variation could lead to a case where one or more of the taps vary by more that 2 LSB. Note: It is recommended to use the PHY ready signal instead of the DFE lock signal in the link layer designs. The DFE lock signal is for testing use only. 0 = DFE is not locked 1 = DFE is locked
11:10	DFE_LOCK_LEN[1:0]	RW 0x3	Decision Feedback Equalizer (DFE) Lock Length. This field indicates the number of DFE updates used when checking for a DFE lock. Used by DFE_LOCK circuit. This control describes the update interval used to test whether DFE is locked. If no significant DFE tap variations (less than 2 LSB variation of all taps) are observed within this update window then DFE is locked. 0 = 256 1 = 512 2 = 1024 3 = 2048
9:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1488: DFE Resolution Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A001C, Port1: 0x000A081C, Port2: 0x000A101C, Port3: 0x000A181C, Port4: 0x000A201C, Port5: 0x000A281C,

Offset Formula: 0x000A001C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
7:4	DFE_STEP_COARSE_FX[3:0]	RW 0x2	Decision Feedback Equalization (DFE) Step Coarse for F1 - F5. This field controls the coarse adaptation step size for the DFE training (fraction of 1 LSB per DFE update). The coarse adjust is used at the beginning of the DFE training to speed up the DFE convergence. 0 = 1/2 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
3:0	DFE_STEP_FINE_FX[3:0]	RW 0x6	Decision Feedback Equalization (DFE) Step Fine for F1 to F5. This field controls the fine adjust step size for the DFE training (fraction of 1 LSB per DFE update). The fine adjust is used at the end of the DFE training 0 = 1/2 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Table 1489: DFE Update Control Register (n=0–6)

Offset: Port0: 0x000A0020, Port1: 0x000A0820, Port2: 0x000A1020, Port3: 0x000A1820, Port4: 0x000A2020, Port5: 0x000A2820,

Offset Formula: 0x000A0020+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	F0_SAT_EN	RW 0x1	F0 Saturate Enable. This field controls if F1-F5 update is disabled when F0 is saturated. The default value disables F1-F5 update when F0 is saturated.
14	F0_SAT_HOLD	RW 0x0	F0 Saturation Hold. This field controls if F0 continues to update after F0 is saturated. The default value continues the F1-F5 update when F0 is saturated.

Copyright © 2015 Marvell

Table 1489: DFE Update Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A0020, Port1: 0x000A0820, Port2: 0x000A1020, Port3: 0x000A1820, Port4: 0x000A2020, Port5: 0x000A2820,

Offset Formula: 0x000A0020+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
13:12	F0_SAT_HOLD_THRES[1:0]	RW 0x2	F0 Saturation Hold Threshold. This field controls the level of the F0 threshold needed to enable the F1-F5 update once F0 gets saturated. 0 = 61 1 = 60 2 = 59 3 = 58
11:6	DFE_UPDATE_EN[5:0]	RW 0x3F	Decision Feedback Equalization (DFE) Update Enable. This field controls whether the DFE taps are allowed to adapt during the DFE training. The taps can be enabled or disabled individually. The MSB is the DFE Tap F5 and the LSB is the DFE Tap F0.
5:0	DFE_FX_FORCE[5:0]	RW 0x0	Decision Feedback Equalization (DFE) FX Force. Each of the six bits has the following options: 0 = Use DFE auto-trained taps. 1 = Use external tap override register values.

Table 1490: DFE F0-F2 Coefficient Control Register (n=0–6)

Offset: Port0: 0x000A0024, Port1: 0x000A0824, Port2: 0x000A1024, Port3: 0x000A1824, Port4: 0x000A2024, Port5: 0x000A2824,

Offset Formula: 0x000A0024+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	DFE_F0_EXT[5:0]	RW 0x20	Decision Feedback Equalization (DFE) F0. Writing to this field sets the external value, if DFE_FX_FORCE is 1.
9:4	DFE_F1_EXT[5:0]	RW 0x0	Decision Feedback Equalization (DFE) F1. Writing to this field sets the external value, if DFE_FX_FORCE is 1.
3:0	DFE_F3_EXT[3:0]	RW 0x0	Decision Feedback Equalization (DFE) F3. Writing to this field sets the external value, if DFE_FX_FORCE is 1.

Table 1491: DFE F3-F5 Coefficient Control Register (n=0–6)

Offset: Port0: 0x000A0028, Port1: 0x000A0828, Port2: 0x000A1028, Port3: 0x000A1828, Port4: 0x000A2028, Port5: 0x000A2828,

Offset Formula: 0x000A0028+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:13	Reserved	RSVD 0x6	Reserved
12:8	DFE_F2_EXT[4:0]	RW 0x0	Decision Feedback Equalization (DFE) F2. Writing to this field sets the external value, if DFE_FX_FORCE is 1.
7:4	DFE_F4_EXT[3:0]	RW 0x0	Decision Feedback Equalization (DFE) F4. Writing to this field sets the external value, if DFE_FX_FORCE is 1.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1491: DFE F3-F5 Coefficient Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A0028, Port1: 0x000A0828, Port2: 0x000A1028, Port3: 0x000A1828, Port4: 0x000A2028, Port5: 0x000A2828,
Offset Formula: 0x000A0028+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	DFE_F5_EXT[3:0]	RW 0x0	Decision Feedback Equalization (DFE) F5. Writing to this field sets the external value, if DFE_FX_FORCE is 1.

Table 1492: Adapted DFE Coefficient 0 Register (n=0–6)
Offset: Port0: 0x000A002C, Port1: 0x000A082C, Port2: 0x000A102C, Port3: 0x000A182C, Port4: 0x000A202C, Port5: 0x000A282C,
Offset Formula: 0x000A002C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:10	DFE_F0[5:0]	RO 0x0	Decision Feedback Equalization (DFE) F0. This field indicates the DFE auto-calibration and auto-trained results.
9:4	DFE_F1[5:0]	RO 0x0	Decision Feedback Equalization (DFE) F1. This field indicates the DFE auto-calibration and auto-trained results.
3:0	DFE_F3[3:0]	RO 0x0	Decision Feedback Equalization (DFE) F0. This field indicates the DFE auto-calibration and auto-trained results.

Table 1493: Adapted DFE Coefficient 1 Register (n=0–6)
Offset: Port0: 0x000A0030, Port1: 0x000A0830, Port2: 0x000A1030, Port3: 0x000A1830, Port4: 0x000A2030, Port5: 0x000A2830,
Offset Formula: 0x000A0030+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:13	ADAPTED_DFE_RES[1:0]	RO 0x0	Adapted DFE resolution
12:8	DFE_F2[4:0]	RO 0x0	Decision Feedback Equalization (DFE) F2. This field indicates the DFE auto-calibration and auto-trained results.
7:4	DFE_F4[3:0]	RO 0x0	Decision Feedback Equalization (DFE) F4. This field indicates the DFE auto-calibration and auto-trained results.
3:0	DFE_F5[3:0]	RO 0x0	Decision Feedback Equalization (DFE) F5. This field indicates the DFE auto-calibration and auto-trained results.

Copyright © 2015 Marvell

Table 1494: Generation 1 Setting 0 Register (n=0–6)

Offset: Port0: 0x000A0034, Port1: 0x000A0834, Port2: 0x000A1034, Port3: 0x000A1834, Port4: 0x000A2034, Port5: 0x000A2834,

Offset Formula: 0x000A0034+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	G1_TX_SLEW_CTRL_EN	RW 0x1	Transmitter Slew Control Enable. This setting is used for PIN_PHY_GEN_TX is 0
14:12	G1_TX_SLEW_RATE_SEL[2:0]	RW 0x4	Transmitter Slew Rate Select. The difference between the slowest and the fastest setting is about 100 ps This setting is used for PIN_PHY_GEN_TX is 0 0 = Fastest edge 7 = Slowest edge
11	G1_TX_EMPH1_EN	RW 0x1	Transmitter Post Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 0
10:7	G1_TX_EMPH1[3:0]	RW 0xA	Transmitter Post Emphasis Value This setting is used for PIN_PHY_GEN_TX is 0
6	G1_TX_AMP_ADJ	RW 0x1	Transmitter Amplitude Adjust. This setting is used for PIN_PHY_GEN_TX is 0 0 = half swing 1 = full swing
5:1	G1_TX_AMP[4:0]	RW 0x16	Transmitter Amplitude. This setting is used for PIN_PHY_GEN_TX is 0
0	G1_TX_SSC_EN	RW 0x0	TX SSC enable

Table 1495: Generation 1 Setting 1 Register (n=0–6)

Offset: Port0: 0x000A0038, Port1: 0x000A0838, Port2: 0x000A1038, Port3: 0x000A1838, Port4: 0x000A2038, Port5: 0x000A2838,

Offset Formula: 0x000A0038+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	G1_TX_AMP_SHFT	RW 0x0	shift TX amplitude 384mv higher when TX_AMP_ADJ is high
14	Reserved	RSVD 0x0	Reserved
13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:11	Reserved	RSVD 0x1	Reserved
10	G1_RX_DFE_EN	RW 0x0	DFE Enable This setting is used for PIN_PHY_GEN_RX is 0

Document Classification: Proprietary Information

Table 1495: Generation 1 Setting 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0038, Port1: 0x000A0838, Port2: 0x000A1038, Port3: 0x000A1838, Port4: 0x000A2038, Port5: 0x000A2838,
Offset Formula: 0x000A0038+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
9:8	G1_RX_SELMUFF[1:0]	RW 0x2	Select Multiple Frequency Final. Sets final coefficient value for receiver CDR frequency loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 0 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)
7:6	G1_RX_SELMUFI[1:0]	RW 0x1	Select Multiple Frequency Initial. Sets the initial bandwidth value for the receiver CDR frequency loop. This bandwidth is used during acquisition. The bandwidth setting can be managed with respect to the data rate. When the training mode is disabled, RX_SETMUFI controls the CDR frequency loop setting, and RX_SELMUFF is not used. RX_SELMUFI and RX_SELMUFF must be programmed to the same values. A smaller number gives a higher loop bandwidth. The ratio between RX_SELMUPF and RX_SELMUFF gives a damping factor. This setting is used for PIN_PHY_GEN_RX is 0 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)
5:3	G1_RX_SELMUPF[2:0]	RW 0x1	Select Multiple Phase Final. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 0 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)

Copyright © 2015 Marvell

Table 1495: Generation 1 Setting 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0038, Port1: 0x000A0838, Port2: 0x000A1038, Port3: 0x000A1838, Port4: 0x000A2038, Port5: 0x000A2838,
Offset Formula: 0x000A0038+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
2:0	G1_RX_SELMUPI[2:0]	RW 0x0	Select Multiple Phase Initial. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 0 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)

Table 1496: Generation 2 Setting 0 Register (n=0–6)
Offset: Port0: 0x000A003C, Port1: 0x000A083C, Port2: 0x000A103C, Port3: 0x000A183C, Port4: 0x000A203C, Port5: 0x000A283C,
Offset Formula: 0x000A003C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	G2_TX_SLEW_CTRL_EN	RW 0x0	Transmitter Slew Control Enable. This setting is used for PIN_PHY_GEN_TX is 1
14:12	G2_TX_SLEW_RATE_SEL[2:0]	RW 0x2	Transmitter Slew Rate Select. The difference between the slowest and the fastest setting is about 100 ps This setting is used for PIN_PHY_GEN_TX is 1 0 = Fastest edge 7 = Slowest edge
11	G2_TX_EMPH1_EN	RW 0x1	Transmitter Post Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 1
10:7	G2_TX_EMPH1[3:0]	RW 0xA	Transmitter Post Emphasis Value This setting is used for PIN_PHY_GEN_TX is 1
6	G2_TX_AMP_ADJ	RW 0x1	Transmitter Amplitude Adjust. This setting is used for PIN_PHY_GEN_TX is 1 0 = half swing 1 = full swing
5:1	G2_TX_AMP[4:0]	RW 0x16	Transmitter Amplitude. This setting is used for PIN_PHY_GEN_TX is 1
0	G2_TX_SSC_EN	RW 0x0	TX SSC enable

Copyright © 2015 Marvell

Table 1497: Generation 2 Setting 1 Register (n=0–6)

Offset: Port0: 0x000A0040, Port1: 0x000A0840, Port2: 0x000A1040, Port3: 0x000A1840, Port4: 0x000A2040, Port5: 0x000A2840,

Offset Formula: 0x000A0040+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	G2_TX_AMP_SHFT	RW 0x0	shift TX amplitude 384mv higher when TX_AMP_ADJ is high
14	Reserved	RSVD 0x0	Reserved
13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:11	Reserved	RSVD 0x1	Reserved
10	G2_RX_DFE_EN	RW 0x0	DFE Enable This setting is used for PIN_PHY_GEN_RX is 1
9:8	G2_RX_SELMUFF[1:0]	RW 0x2	Select Multiple Frequency Final. Sets final coefficient value for receiver CDR frequency loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 1 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)
7:6	G2_RX_SELMUFI[1:0]	RW 0x1	Select Multiple Frequency Initial. Sets the initial bandwidth value for the receiver CDR frequency loop. This bandwidth is used during acquisition. The bandwidth setting can be managed with respect to the data rate. When the training mode is disabled, RX_SETMUFI controls the CDR frequency loop setting, and RX_SELMUFF is not used. RX_SELMUFI and RX_SELMUFF must be programmed to the same values. A smaller number gives a higher loop bandwidth. The ratio between RX_SELMUPF and RX_SELMUFF gives a damping factor. This setting is used for PIN_PHY_GEN_RX is 1 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)

Copyright © 2015 Marvell

Table 1497: Generation 2 Setting 1 Register (n=0–6) (Continued)

Offset: Port0: 0x000A0040, Port1: 0x000A0840, Port2: 0x000A1040, Port3: 0x000A1840, Port4: 0x000A2040, Port5: 0x000A2840,

Offset Formula: 0x000A0040+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
5:3	G2_RX_SELMUPF[2:0]	RW 0x1	Select Multiple Phase Final. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 1 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)
2:0	G2_RX_SELMUPI[2:0]	RW 0x0	Select Multiple Phase Initial. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 1 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)

Table 1498: Generation 3 Setting 0 Register (n=0–6)

Offset: Port0: 0x000A0044, Port1: 0x000A0844, Port2: 0x000A1044, Port3: 0x000A1844, Port4: 0x000A2044, Port5: 0x000A2844,

Offset Formula: 0x000A0044+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	G3_TX_SLEW_CTRL_EN	RW 0x0	Transmitter Slew Control Enable. This setting is used for PIN_PHY_GEN_TX is 2
14:12	G3_TX_SLEW_RATE_SEL[2:0]	RW 0x0	Transmitter Slew Rate Select. The difference between the slowest and the fastest setting is about 100 ps This setting is used for PIN_PHY_GEN_TX is 2 0 = Fastest edge 7 = Slowest edge
11	G3_TX_EMPH1_EN	RW 0x1	Transmitter Post Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 2
10:7	G3_TX_EMPH1[3:0]	RW 0x6	Transmitter Post Emphasis Value This setting is used for PIN_PHY_GEN_TX is 2
6	G3_TX_AMP_ADJ	RW 0x1	Transmitter Amplitude Adjust. This setting is used for PIN_PHY_GEN_TX is 2 0 = half swing 1 = full swing

Copyright © 2015 Marvell

Table 1498: Generation 3 Setting 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0044, Port1: 0x000A0844, Port2: 0x000A1044, Port3: 0x000A1844, Port4: 0x000A2044, Port5: 0x000A2844,
Offset Formula: 0x000A0044+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
5:1	G3_TX_AMP[4:0]	RW 0x10	Transmitter Amplitude. This setting is used for PIN_PHY_GEN_TX is 2
0	G3_TX_SSC_EN	RW 0x0	TX SSC enable

Table 1499: Generation 3 Setting 1 Register (n=0–6)
Offset: Port0: 0x000A0048, Port1: 0x000A0848, Port2: 0x000A1048, Port3: 0x000A1848, Port4: 0x000A2048, Port5: 0x000A2848,
Offset Formula: 0x000A0048+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	G3_TX_AMP_SHFT	RW 0x0	shift TX amplitude 384mv higher when TX_AMP_ADJ is high
14	Reserved	RSVD 0x1	Reserved
13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:11	Reserved	RSVD 0x2	Reserved
10	G3_RX_DFE_EN	RW 0x0	DFE Enable This setting is used for PIN_PHY_GEN_RX is 2
9:8	G3_RX_SELMUFF[1:0]	RW 0x3	Select Multiple Frequency Final. Sets final coefficient value for receiver CDR frequency loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 2 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)

Copyright © 2015 Marvell

Table 1499: Generation 3 Setting 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0048, Port1: 0x000A0848, Port2: 0x000A1048, Port3: 0x000A1848, Port4: 0x000A2048, Port5: 0x000A2848,
Offset Formula: 0x000A0048+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
7:6	G3_RX_SELMUFI[1:0]	RW 0x2	Select Multiple Frequency Initial. Sets the initial bandwidth value for the receiver CDR frequency loop. This bandwidth is used during acquisition. The bandwidth setting can be managed with respect to the data rate. When the training mode is disabled, RX_SETMUFI controls the CDR frequency loop setting, and RX_SELMUFF is not used. RX_SELMUFI and RX_SELMUFF must be programmed to the same values. A smaller number gives a higher loop bandwidth. The ratio between RX_SELMUPF and RX_SELMUFF gives a damping factor. This setting is used for PIN_PHY_GEN_RX is 2 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)
5:3	G3_RX_SELMUPF[2:0]	RW 0x2	Select Multiple Phase Final. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 2 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)
2:0	G3_RX_SELMUPI[2:0]	RW 0x1	Select Multiple Phase Initial. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 2 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)

Copyright © 2015 Marvell

Table 1500: Generation 4 Setting 0 Register (n=0–6)

Offset: Port0: 0x000A004C, Port1: 0x000A084C, Port2: 0x000A104C, Port3: 0x000A184C, Port4: 0x000A204C, Port5: 0x000A284C,

Offset Formula: 0x000A004C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	G4_TX_SLEW_CTRL_EN	RW 0x0	Transmitter Slew Control Enable. This setting is used for PIN_PHY_GEN_TX is 3
14:12	G4_TX_SLEW_RATE_SEL[2:0]	RW 0x0	Transmitter Slew Rate Select. The difference between the slowest and the fastest setting is about 100 ps This setting is used for PIN_PHY_GEN_TX is 3 0 = Fastest edge 7 = Slowest edge
11	G4_TX_EMPH1_EN	RW 0x1	Transmitter Post Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 3
10:7	G4_TX_EMPH1[3:0]	RW 0x6	Transmitter Post Emphasis Value This setting is used for PIN_PHY_GEN_TX is 3
6	G4_TX_AMP_ADJ	RW 0x1	Transmitter Amplitude Adjust. This setting is used for PIN_PHY_GEN_TX is 3 0 = half swing 1 = full swing
5:1	G4_TX_AMP[4:0]	RW 0x10	Transmitter Amplitude. This setting is used for PIN_PHY_GEN_TX is 3
0	G4_TX_SSC_EN	RW 0x0	TX SSC enable

Table 1501: Generation 4 Setting 1 Register (n=0–6)

Offset: Port0: 0x000A0050, Port1: 0x000A0850, Port2: 0x000A1050, Port3: 0x000A1850, Port4: 0x000A2050, Port5: 0x000A2850,

Offset Formula: 0x000A0050+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	G4_TX_AMP_SHFT	RW 0x0	shift TX amplitude 384mv higher when TX_AMP_ADJ is high
14	Reserved	RSVD 0x1	Reserved
13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:11	Reserved	RSVD 0x2	Reserved

Copyright © 2015 Marvell

Table 1501: Generation 4 Setting 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0050, Port1: 0x000A0850, Port2: 0x000A1050, Port3: 0x000A1850, Port4: 0x000A2050, Port5: 0x000A2850,
Offset Formula: 0x000A0050+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
10	G4_RX_DFE_EN	RW 0x0	DFE Enable This setting is used for PIN_PHY_GEN_RX is 3 This setting is used for PIN_PHY_GEN_RX is 3
9:8	G4_RX_SELMUFF[1:0]	RW 0x3	Select Multiple Frequency Final. Sets final coefficient value for receiver CDR frequency loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 3 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)
7:6	G4_RX_SELMUFI[1:0]	RW 0x2	Select Multiple Frequency Initial. Sets the initial bandwidth value for the receiver CDR frequency loop. This bandwidth is used during acquisition. The bandwidth setting can be managed with respect to the data rate. When the training mode is disabled, RX_SETMUFI controls the CDR frequency loop setting, and RX_SELMUFF is not used. RX_SELMUFI and RX_SELMUFF must be programmed to the same values. A smaller number gives a higher loop bandwidth. The ratio between RX_SELMUPF and RX_SELMUFF gives a damping factor. This setting is used for PIN_PHY_GEN_RX is 3 0 = 1/256 (largest bandwidth) 1 = 1/512 2 = 1/1024 3 = 1/2048 (smallest bandwidth)
5:3	G4_RX_SELMUPF[2:0]	RW 0x2	Select Multiple Phase Final. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 0 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)

Copyright © 2015 Marvell

Table 1501: Generation 4 Setting 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0050, Port1: 0x000A0850, Port2: 0x000A1050, Port3: 0x000A1850, Port4: 0x000A2050, Port5: 0x000A2850,
Offset Formula: 0x000A0050+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
2:0	G4_RX_SELMUPI[2:0]	RW 0x1	Select Multiple Phase Initial. Set Initial Coefficient Value for Receiver CDR Phase Loop. The bandwidth setting can be managed with respect to the data rate. This setting is used for PIN_PHY_GEN_RX is 3 0 = 1 1 = 1/2 2 = 1/4 3 = 1/8 (smallest bandwidth) 4 = 2 5 = 4 (largest bandwidth)

Table 1502: PHY Test Control 0 Register (n=0–6)
Offset: Port0: 0x000A0054, Port1: 0x000A0854, Port2: 0x000A1054, Port3: 0x000A1854, Port4: 0x000A2054, Port5: 0x000A2854,
Offset Formula: 0x000A0054+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	PT_EN	RW 0x0	PHY Test Enable Pattern should be selected while PT_EN = 0. When set to 1, this signal clears the pattern and error counts and begins looking for pattern lock. When set to zero this stops the testing and freeze the error and pattern counts. 0 = PHY Test disable 1 = PHY Test enable
14	PT_RST	RW 0x0	PHY Test Reset. This is a reset signal to PHY Test. Once its set to 1, all registers in PHY Test are cleared.
13	PT_PASS	RO 0x0	PHY Test Pass 0 = 1 or more errors or the pattern is not locked 1 = Pattern is locked and no errors detected.
12	PT_LOCK	RO 0x0	PHY Test Pattern Lock If the pattern doesn’t lock then either the signal quality is low or the pattern provided to the receiver doesn’t match the programmed pattern. 0 = Pattern detector is not locked onto the pattern 1 = Patter detector had locked onto the pattern

Copyright © 2015 Marvell

Table 1502: PHY Test Control 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0054, Port1: 0x000A0854, Port2: 0x000A1054, Port3: 0x000A1854, Port4: 0x000A2054, Port5: 0x000A2854,
Offset Formula: 0x000A0054+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11	PT_XFER_DIFF	RW 0x0	PHY Test Transmit Jitter Pattern Select Option When this signal is 1 the transmitter and receiver can be programmed to use different jitter patterns. This field only has a meaning when using the jitter patterns PT_PATTERN_SEL = 1110b. Both fields PT_DATA [95:88] and PT_DATA [7:0] have the same encoding. 0 = PT_DATA[7:0] select jitter test pattern for both transmit and receive data 1 = PT_DATA [95:88] selects jitter test pattern for transmit data and PT_DATA [7:0] selects pattern for receive data comparison
10	PT_PATSYN_EN	RW 0x0	PHY Test Pattern Sync Enable If enabled, a pattern sync signal is generated and pulsed at the pattern repeat point. This can be used as a pattern trigger for test equipment.
9	PT_START_RD	RW 0x0	PHY Test Start Running Disparity Selects the initial running disparity for SATA test patterns. 0 = Initial Disparity for pattern is negative 1 = Initial Disparity for pattern is positive
8	PT_LONG_SHORT	RW 0x0	PHY Test Long SATA Patterns 0 = Short version of SATA patterns 1 = Long version of SATA patterns
7:4	PT_PATTERN_SEL[3:0]	RW 0x0	PHY Test Pattern Select This selects the pattern from the table of patterns or specifies an alternate table to use for additional jitter test patterns. Patterns can only be selected when PT_EN = 0.
3:0	PT_LOCK_CNT[3:0]	RW 0x0	Pattern Lock Count Program with the number of pattern signatures to be found -1. Once n + 1 of these signatures has been found PT_LOCK is asserted and the bit error test begins.

Table 1503: PHY Test Data 0 Register (n=0–6)
Offset: Port0: 0x000A0058, Port1: 0x000A0858, Port2: 0x000A1058, Port3: 0x000A1858, Port4: 0x000A2058, Port5: 0x000A2858,
Offset Formula: 0x000A0058+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1503: PHY Test Data 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0058, Port1: 0x000A0858, Port2: 0x000A1058, Port3: 0x000A1858, Port4: 0x000A2058, Port5: 0x000A2858,
Offset Formula: 0x000A0058+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_DATA[95:80]	RW 0x0	PHY Test Data[95:80] When using Jitter test patterns PT_PATTERN_SEL = 1110b and PT_XFER_DIFF = 1, the transmit pattern is selected from PT_DATA[95:88] instead of using PT_DATA[7:0]. In this case the receive pattern is still selected using PT_DATA[7:0] If using jitter test pattern 5 PT_PATTERN_SEL = 1110b and PT_DATA[7:0] = 00000101b then PT_DATA[87:8] = an 80b user defined pattern to transmit and receive without encoding/decoding.

Table 1504: PHY Test Data 1 Register (n=0–6)
Offset: Port0: 0x000A005C, Port1: 0x000A085C, Port2: 0x000A105C, Port3: 0x000A185C, Port4: 0x000A205C, Port5: 0x000A285C,
Offset Formula: 0x000A005C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_DATA[79:64]	RW 0x0	PHY Test Data[79:64] If using jitter test pattern 5, PT_PATTERN_SEL = 1110b and PT_DATA[7:0] = 00000101b then PT_DATA[87:8] = an 80 b user defined pattern to transmit and receive without encoding/decoding.

Table 1505: PHY Test Data 2 Register (n=0–6)
Offset: Port0: 0x000A0060, Port1: 0x000A0860, Port2: 0x000A1060, Port3: 0x000A1860, Port4: 0x000A2060, Port5: 0x000A2860,
Offset Formula: 0x000A0060+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_DATA[63:48]	RW 0x0	PHY Test Data[63:48] If PT_PATTERN_SEL = 1111b then PT_DATA[63:0] specifies 8-bit symbols to encoded and transmit with 8B10B encoding. If using jitter test pattern 5 PT_PATTERN_SEL = 1110b , and PT_DATA[7:0] = 00000101b then PT_DATA[87:8] = an 80b user defined pattern to transmit and receive without encoding/decoding.

Copyright © 2015 Marvell

Table 1506: PHY Test Data 3 Register (n=0–6)

Offset: Port0: 0x000A0064, Port1: 0x000A0864, Port2: 0x000A1064, Port3: 0x000A1864, Port4: 0x000A2064, Port5: 0x000A2864,

Offset Formula: 0x000A0064+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	PT_DATA[47:32]	RW 0x0	PHY Test Data[47:32] If PT_PATTERN_SEL = 1111b then PT_DATA[63:0] specifies 8-bit symbols to encoded and transmit with 8B10B encoding. If using jitter test pattern 5, PT_PATTERN_SEL = 1110b and PT_DATA[7:0] = 00000101b then PT_DATA[87:8] = an 80b user defined pattern to transmit and receive without encoding/decoding.

Table 1507: PHY Test Data 4 Register (n=0–6)

Offset: Port0: 0x000A0068, Port1: 0x000A0868, Port2: 0x000A1068, Port3: 0x000A1868, Port4: 0x000A2068, Port5: 0x000A2868,

Offset Formula: 0x000A0068+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	PT_DATA[31:16]	RW 0x0	PHY Test Data[31:16] If PT_PATTERN_SEL = 1111b then PT_DATA[63:0] specifies 8-bit symbols to encoded and transmit with 8B10B encoding. If using jitter test pattern 5, PT_PATTERN_SEL = 1110b and PT_DATA[7:0] = 00000101b then PT_DATA[87:8] = an 80b user defined pattern to transmit and receive without encoding/decoding.

Copyright © 2015 Marvell

Table 1508: PHY Test Data 5 Register (n=0–6)

Offset: Port0: 0x000A006C, Port1: 0x000A086C, Port2: 0x000A106C, Port3: 0x000A186C, Port4: 0x000A206C, Port5: 0x000A286C,

Offset Formula: 0x000A006C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_DATA[15:0]	RW 0x0	PHY Test Data[15:0] When using jitter test patterns (PT_PATTERN_SEL = 1110b) the jitter pattern is selected using bits PHY Test Data[7:0]. If PT_PATTERN_SEL = 1111b then PT_DATA[63:0] specifies 8-bit symbols to encode and transmit with 8B10B encoding. If using jitter test pattern 5 PT_PATTERN_SEL = 1110b and PT_DATA[7:0] = 00000101b then PT_DATA[87:8] = an 80b user defined pattern to transmit and receive without encoding/decoding.

Table 1509: PHY Test PRBS Counter 0 Register (n=0–6)

Offset: Port0: 0x000A0070, Port1: 0x000A0870, Port2: 0x000A1070, Port3: 0x000A1870, Port4: 0x000A2070, Port5: 0x000A2870,

Offset Formula: 0x000A0070+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_PRBS_CNT[47:32]	RO 0x0	PHY Test Pattern Count [47:32] The number of 40-bit patterns received since acquiring pattern lock.

Table 1510: PHY Test PRBS Counter 1 Register (n=0–6)

Offset: Port0: 0x000A0074, Port1: 0x000A0874, Port2: 0x000A1074, Port3: 0x000A1874, Port4: 0x000A2074, Port5: 0x000A2874,

Offset Formula: 0x000A0074+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_PRBS_CNT[31:16]	RO 0x0	PHY Test Pattern Count [31:16] The number of 40-bit patterns received since acquiring pattern lock.

Table 1511: PHY Test PRBS Counter 2 Register (n=0–6)

Offset: Port0: 0x000A0078, Port1: 0x000A0878, Port2: 0x000A1078, Port3: 0x000A1878, Port4: 0x000A2078, Port5: 0x000A2878,

Offset Formula: 0x000A0078+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	PT_PRBS_CNT[15:0]	RO 0x0	PHY Test Pattern Count [15:0] The number of 40-bit patterns received since acquiring pattern lock.

Copyright © 2015 Marvell

Table 1512: PHY Test PRBS Error Counter 0 Register (n=0–6)

Offset: Port0: 0x000A007C, Port1: 0x000A087C, Port2: 0x000A107C, Port3: 0x000A187C, Port4: 0x000A207C, Port5: 0x000A287C,

Offset Formula: 0x000A007C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_PRBS_ERR_CNT[31:16]	RO 0x0	PHY Test Error Count [31:16] Indicates how many bit errors were encountered after obtaining pattern lock.

Table 1513: PHY Test PRBS Error Counter 1 Register (n=0–6)

Offset: Port0: 0x000A0080, Port1: 0x000A0880, Port2: 0x000A1080, Port3: 0x000A1880, Port4: 0x000A2080, Port5: 0x000A2880,

Offset Formula: 0x000A0080+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	PT_PRBS_ERR_CNT[15:0]	RO 0x0	PHY Test Error Count [15:0] The error count indicates how many bit errors were encountered after obtaining pattern lock.

Table 1514: PHY Test OOB 0 Register (n=0–6)

Offset: Port0: 0x000A0084, Port1: 0x000A0884, Port2: 0x000A1084, Port3: 0x000A1884, Port4: 0x000A2084, Port5: 0x000A2884,

Offset Formula: 0x000A0084+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:15	Reserved	RSVD 0x0	Reserved
14	PT_PHYREADY_FORCE	RW 0x0	PHY Test PHY Ready Force. Forces the PHY Ready in PHY Test to be 1.
13:12	PT_TESTMODE[1:0]	RW 0x0	Test Mode. 0 = old test mode (as R1.0) which takes PT_EN from register 1 = respect I_PHYREADY 2 = respect REG_PHYREADY 3 = reserved, do not use
11:0	Reserved	RSVD 0x30	Reserved

Table 1515: PHY Test OOB Register (n=0–6)

Offset: Port0: 0x000A0088, Port1: 0x000A0888, Port2: 0x000A1088, Port3: 0x000A1888, Port4: 0x000A2088, Port5: 0x000A2888,

Offset Formula: 0x000A0088+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1000	Reserved

Copyright © 2015 Marvell

Table 1516: Digital Loopback Enable Register (n=0–6)
Offset: Port0: 0x000A008C, Port1: 0x000A088C, Port2: 0x000A108C, Port3: 0x000A188C, Port4: 0x000A208C, Port5: 0x000A288C,
Offset Formula: 0x000A008C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	LOCAL_DIG_RX2TX_LPBK_EN	RW 0x0	Far End Parallel Loopback Enable Receiver to Transmitter. This mode does not tolerate the link-rate differences and does not insert or delete characters for the link-rate tolerance. If the data driven into the receiver is driven off by a clock that is synchronous to the PHY reference clock, then the use of RXPHER_TO_TX_EN is optional. This loopback mode includes a FIFO, which is used to absorb the jitter on the received signal when RXPHER_TO_TX_EN = 0. If RXPHER_TO_TX_EN is set to 0, then the transmitted data is re-timed using the transmit clock. If the received data is from a asynchronous clock domain compared to the PHY reference clock, then RXPHER_TO_TX_EN should be set to 1. For this case, the receive clock is used to time the transmitted data and the jitter tracked by the recovered clock is present in the transmitted data. 0 = No loopback enabled. 1 = Send received data back out the transmitter.
14	LOCAL_DIG_TX2RX_LPBK_EN	RW 0x0	Parallel Transmit to Receive Loopback. The parallel data received on the transmit data bus for the transmission is looped back to the receive parallel bus and bypasses the majority of the PHY circuits.
13	LOCAL_ANA_TX2RX_LPBK_EN	RW 0x0	Local Analog Transmit to Receive Loop Back Enable. The transmit data from the pre-driver is looped back to the serial sampler. This loopback mode bypasses the last state of the transmit driver as well as the receiver Squelch detector and Feed Forward Equalization. In this mode, the Squelch detect signal (sq_det) should not be used by the link layer logic.
12	RXPHER_TO_TX_EN	RW 0x0	Use the receiver's recovered clock to drive the transmit data. When set to 1, jitter that is tracked by the recovered clock is transferred to the transmitted data stream. Use RX DPHER to control TX intp to make RXDCLK sync with TXDCLK. 0 = Transmit data is timed using the transmit clock 1 = Transmit data is timed using the receive clock
11	PT_CNT_RST	RW 0x0	PHY Test Pattern counter reset
10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:7	Reserved	RSVD 0x0	Reserved
6	NORMAL_STATE_READY	RO 0x0	Normal State Ready. Same function as PIN_RX_INIT_DONE. 0 = Receiver initialization state machine is not in the normal operating state 1 = Receiver initialization state machine is in the normal operating (tracking) state.

Copyright © 2015 Marvell

Table 1516: Digital Loopback Enable Register (n=0–6) (Continued)
Offset: Port0: 0x000A008C, Port1: 0x000A088C, Port2: 0x000A108C, Port3: 0x000A188C, Port4: 0x000A208C, Port5: 0x000A288C,
Offset Formula: 0x000A008C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
5	PLL_READY_RX	RO 0x0	Register for PIN_PLL_READY_RX
4	PLL_READY_TX	RO 0x0	Register for PIN_PLL_READY_TX
3:1	SEL_BITS[2:0]	RW 0x1	Select Data Bit Width 0 = Reserved 1 = 20-bit 2 = 40-bit 3 = Reserved 4 = Reserved 5 = 16-bit 6 = 32-bit 7 = Reserved
0	Reserved	RSVD 0x0	Reserved

Table 1517: Sync Pattern Register (n=0–6)
Offset: Port0: 0x000A0090, Port1: 0x000A0890, Port2: 0x000A1090, Port3: 0x000A1890, Port4: 0x000A2090, Port5: 0x000A2890,
Offset Formula: 0x000A0090+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	SYNC_DET_EN	RW 0x0	Sync Detect Enable.
14	SYNC_POL	RW 0x1	Sync Polarity. It is sync character detection mode. 0 = SYNC_CHAR only 1 = SYNC_CHAR or inverted SYNC_CHAR.
13	SYNC_FOUND_LEVEL	RO 0x0	Sync Found Level. Sync Found Level. This is a latched version of sync detected. Toggle ALIGN_STAT_RD_REQ from 0 to 1 to 0. This sets this bit to 0.
12	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1517: Sync Pattern Register (n=0–6) (Continued)

Offset: Port0: 0x000A0090, Port1: 0x000A0890, Port2: 0x000A1090, Port3: 0x000A1890, Port4: 0x000A2090, Port5: 0x000A2890,

Offset Formula: 0x000A0090+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11	RXD_INV	RW 0x0	Receive Polarity Invert. It is receive polarity swap enable. This control allows PCB to select a layout where the plus and minus signals are swapped. Internally the logic can invert these bits. This inversion comes before the PHY Test logic so that the PHY Test module sees the inversion. 0 = No polarity swap 1 = Polarity Swap (1 = 0 and 0 = 1)
10	TXD_INV	RW 0x0	Transmit Polarity Invert. It is transmit polarity swap enable. This control allows PCB to select a layout where the plus and minus signals are swapped. Internally the logic can invert these bits. This inversion comes after PHY Test so that the PHY Test patterns would also be inverted. 0 = No polarity swap 1 = Polarity Swap (1 = 0 and 0 = 1)
9:0	SYNC_CHAR[9:0]	RW 0x283	Sync Character Pattern This field is used to program the sync detector circuit with the pattern to match when checking for 10-bit character sync. The SYNC_MASK field indicates which of these bits are used in the comparison.

Table 1518: Sync Mask and Maximal Generation Setting Register (n=0–6)

Offset: Port0: 0x000A0094, Port1: 0x000A0894, Port2: 0x000A1094, Port3: 0x000A1894, Port4: 0x000A2094, Port5: 0x000A2894,

Offset Formula: 0x000A0094+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x0	Reserved
14:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12	DET_BYPASS	RW 0x1	byass frame detection and sync detection 0 = use frame detection or sync detection result, PIN_RXDATA has cycle delay from analog, if sync detection is used or frame marker detection is used, this bit must be set to 0 1 = bypass analog data, PIN_RXDATA has no delay from analog

Table 1518: Sync Mask and Maximal Generation Setting Register (n=0–6) (Continued)
Offset: Port0: 0x000A0094, Port1: 0x000A0894, Port2: 0x000A1094, Port3: 0x000A1894, Port4: 0x000A2094, Port5: 0x000A2894,
Offset Formula: 0x000A0094+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11:10	PHY_GEN_MAX[1:0]	RW 0x2	Maximal PHY Generation Setting User should set this register to specify the maximal generation for PHY to support For SerDes: Don't care Note: Binary settings not mentioned here are not to be used. 0 = 1.5 Gbps 0 = 1.0625 Gbps 0 = 2.5 Gbps 0 = 1.5 Gbps 1 = 5 Gbps 1 = 5 Gbps 1 = 3 Gbps 1 = 2.125 Gbps 1 = 3 Gbps 2 = 6 Gbps 2 = 8 Gbps 2 = 4.25 Gbps 2 = 6 Gbps
9:0	SYNC_MASK[9:0]	RW 0x3FF	Sync character mask Each bit that is a 1 is compared to the SYNC_CHAR for a pattern match. Each bit that is a 0 is ignored during the sync detect comparison.

Table 1519: PHY Isolation Mode Control Register (n=0–6)
Offset: Port0: 0x000A0098, Port1: 0x000A0898, Port2: 0x000A1098, Port3: 0x000A1898, Port4: 0x000A2098, Port5: 0x000A2898,
Offset Formula: 0x000A0098+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	PHY_ISOLATE_MODE	RW 0x0	PHY Isolate Mode 0 = Normal mode 1 = PHY isolate mode (used to isolate the PHY from outside logic to test in stand alone mode)
14	Reserved	RSVD 0x0	Reserved
13	REFCLK_ON_DCLK_DIS	RW 0x0	Referenc Clock on TXDCLK/RXDCLK Disable 0 = Enable reference clock on PIN_TXDCLK and PIN_RXDCLK when PIN_PLL_READY is low 1 = Disable reference clock on PIN_TXDCLK and PIN_RXDCLK when PIN_PLL_READY is low
12	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1519: PHY Isolation Mode Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A0098, Port1: 0x000A0898, Port2: 0x000A1098, Port3: 0x000A1898, Port4: 0x000A2098, Port5: 0x000A2898,
Offset Formula: 0x000A0098+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11	AUTO_RX_INIT_DIS	RW 0x1	Automatic rx_init Disable 0 = Enable automatic rx_init trigger after power up, partial slumber/slumber and speed change 1 = Disable automatic rx_init trigger after power up, partial slumber/slumber and speed change
10	RX_INIT	RW 0x0	Receiver Initialization.
9	Reserved	RSVD 0x0	Reserved
8	TX_DRV_IDLE	RW 0x1	Transmitter Driver Idle. The common mode voltage varies by the transmitter amplitude setting. 0 = Driver output valid 1 = Driver at common mode voltage
7:4	PHY_GEN_TX[3:0]	RW 0x2	Transmitter Generation Select. Note: Binary settings not mentioned here are not to be used. 0 = 2.5 Gbps 0 = Reserved 0 = 1.0625 Gbps 0 = 1.5 Gbps 0 = 1.5 Gbps 0 = 1.5 Gbps 1 = 5 Gbps 1 = 5 Gbps 1 = 3 Gbps 1 = 3 Gbps 1 = 2.125 Gbps 1 = 3 Gbps 2 = 6 Gbps 2 = 6 Gbps 2 = 8 Gbps 2 = 6 Gbps 2 = 4.25 Gbps 3 = 5 Gbps 4 = 4 Gbps 5 = 1.125 Gbps 6 = 1.25 Gbps 7 = 2.5 Gbps 8 = 3.125 Gbps 9 = 3.75 Gbps 10 = VA: 4.25 Gbps 11 = VB: 6.25 Gbps 12 = VC: 5.20833 Gpbs 13 = VD: 7.5 Gbps

Copyright © 2015 Marvell

Table 1519: PHY Isolation Mode Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A0098, Port1: 0x000A0898, Port2: 0x000A1098, Port3: 0x000A1898, Port4: 0x000A2098, Port5: 0x000A2898,
Offset Formula: 0x000A0098+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	PHY_GEN_RX[3:0]	RW 0x2	Receiver Generation Select. Note: Binary settings not mentioned here are not to be used. 0 = 2.5 Gbps 0 = Reserved 0 = 1.0625 Gbps 0 = 1.5 Gbps 0 = 1.5 Gbps 0 = 1.5 Gbps 1 = 5 Gbps 1 = 5 Gbps 1 = 3 Gbps 1 = 3 Gbps 1 = 2.125 Gbps 1 = 3 Gbps 2 = 6 Gbps 2 = 6 Gbps 2 = 8 Gbps 2 = 6 Gbps 2 = 4.25 Gbps 3 = 5 Gbps 4 = 4 Gbps 5 = 1.125 Gbps 6 = 1.25 Gbps 7 = 2.5 Gbps 8 = 3.125 Gbps 9 = 3.75 Gbps 10 = VA: 4.25 Gbps 11 = VB: 6.25 Gbps 12 = VC: 5.20833 Gpbs 13 = VD: 7.5 Gbps

Table 1520: EOM Circuit Control Register (n=0–6)
Offset: Port0: 0x000A009C, Port1: 0x000A089C, Port2: 0x000A109C, Port3: 0x000A189C, Port4: 0x000A209C, Port5: 0x000A289C,
Offset Formula: 0x000A009C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	EOM_EN	RW 0x0	Eye-opening Measurement (EOM) Circuit Enable Signal. 0 = EOM is disabled 1 = EOM is enabled
14	Reserved	RSVD 0x0	Reserved
13	EOM_READY	RO 0x0	EOM Ready 0 = Receiver state machine has not passed the control to the EOM. 1 = Receiver state machine has passed the control to the EOM.

Document Classification: Proprietary Information

Table 1520: EOM Circuit Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A009C, Port1: 0x000A089C, Port2: 0x000A109C, Port3: 0x000A189C, Port4: 0x000A209C, Port5: 0x000A289C,
Offset Formula: 0x000A009C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	EOM_MEASURE_STATE[1:0]	RW 0x0	EOM Measure State.
9:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:4	EOM_SAM_VLD_TIME[2:0]	RW 0x0	Eye-opening Measurement (EOM) Valid Time. This field sets the hardware waiting period before the EOM measurement starts. others: 27 ns 0 = 500 ns 1 = 700 ns 2 = 1000 ns 3 = 1500 ns 4 = 2000 ns
3:1	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
0	Reserved	RSVD 0x0	Reserved

Table 1521: EOM Set 1 Control 0 Register (n=0–6)
Offset: Port0: 0x000A00A0, Port1: 0x000A08A0, Port2: 0x000A10A0, Port3: 0x000A18A0, Port4: 0x000A20A0, Port5: 0x000A28A0,
Offset Formula: 0x000A00A0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	EOM_SET1_STOP_EN[1:0]	RW 0x0	Eye-opening Measurement (EOM) Stop Enable. This field is used to program the stop conditions. 0 = Stop only if the population limit is reached. 1 = Force the EOM Set1 measurement to stop. 2 = Stop if the error threshold is reached. 3 = Force the EOM Set1 measurement to stop.
13	EOM_SET1_VLD	RW 0x0	Eye-opening Measurement (EOM) Valid. Set to 1 to indicate the EOM configuration is valid and to initiate the measurement. If a set 2 EOM measurement is running when EOM_SET1_VLD is set to 1, the set1 measurement does not begin until the set 2 measurement stops. EOM_SET1_VLD must be cleared manually after reading the EOM_SET1_STOP_STS, EOM_SET1_SAMPLE_CNT[63:0], EOM_SET1_ERROR_CNT[63:0]. These fields are cleared when EOM_SET1_VLD is programmed from 0 to 1.

Copyright © 2015 Marvell

Table 1521: EOM Set 1 Control 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A00A0, Port1: 0x000A08A0, Port2: 0x000A10A0, Port3: 0x000A18A0, Port4: 0x000A20A0, Port5: 0x000A28A0,
Offset Formula: 0x000A00A0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
12:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:6	EOM_SET1_SIGN_BIT[1:0]	RW 0x0	EOM Sign Bits This control is used to specify what bit combinations to sample and which half of the eye diagram is to be measured (top or bottom). The LSB of this field indicates if EOM is measuring the top half or bottom half of the eye. LSB If the MSB is opposite of the LSB, then EOM is measuring transition bits. If the MSB is the same as the LSB, then EOM is measuring non-transition bits. To measure the error rate at a specific voltage and phase, the EOM software must measure transition and non-transition bits separately and combined the results. If there is not an equal number of transition and non-transition bits in the pattern, then the EOM measurement would need to take into account how to combine the transition errors and non-transition errors. One special case is for the 101010 clock like pattern. In this case, only transition bits should be measured since there are no non-transition bits. 0 = Measure non-transition bits on the bottom half of the eye. 0 = Bottom half 1 = Measure transition bits on the top half of the eye. 1 = Top half 2 = Measure transition bits on the bottom half of the eye. 3 = Measure non-transition bits on the top half of the eye.
5:0	EOM_SET1_VOLTAGE[5:0]	RW 0x0	EOM Voltage. This unsigned value indicates what threshold voltage to use for the error comparator. This value is multiplied by the DFE_RES setting which results in a threshold voltage of 0 - 63 x DFE_RES. This range is used for half the eye's vertical range; therefore, the EOM eye does not have twice the range (-63/DFE_RES to 63/DFE_RES). If the DFE taps F1-F5 are not all zeros then DFE_RES should not be changed because this would alter the signal's equalization in addition to the vertical range DFE_RES Range 0 = 5mv to 315mv 1 = 4mv to 252mv 2 = 3.3mv to 207.9mv 3 = 2.9mv to 182.7mv

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1522: EOM Set 1 Control 1 Register (n=0–6)

Offset: Port0: 0x000A00A4, Port1: 0x000A08A4, Port2: 0x000A10A4, Port3: 0x000A18A4, Port4: 0x000A20A4, Port5: 0x000A28A4,

Offset Formula: 0x000A00A4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	EOM_SET1_RDY	RO 0x0	Eye-opening Measurement (EOM) Ready. When the SET1 measurement stops, this bit is set to 1 to indicate that the measurement is complete and that the stop status, error counter, and sample counter can be accessed.
14:12	EOM_SET1_STOP_STS[2:0]	RO 0x0	Status of stop due to population. Bit[2] is 1 when sample count is equal to population count. Bit[1] is 1 when error count meets the threshold register. Bit[0] is 1 when SW requests a stop. These bits are valid after EOM_SET1_RDY = 1.
11:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	EOM_SET1_PHASE[6:0]	RW 0x0	Eye-opening Measurement (EOM) Phase. This field controls the sampling phase for the error comparator. The resolution depends on the bit rate. For the data rate of 6 Gbps, the resolution is 1/64 of a unit interval. For the data rate of 4 Gbps/3 Gbps/2 Gbps/1.5 Gbps/1 Gbps, the resolution is 1/128 of a unit interval. The phase adjustments must be limited to less than or equal to steps of 4 to avoid glitches in internal clocks.

Table 1523: EOM Set 1 Population Counter 0 Register (n=0–6)

Offset: Port0: 0x000A00A8, Port1: 0x000A08A8, Port2: 0x000A10A8, Port3: 0x000A18A8, Port4: 0x000A20A8, Port5: 0x000A28A8,

Offset Formula: 0x000A00A8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	EOM_SET1_POP_CNT[63:48]	RW 0xFFFF	Eye-opening Measurement (EOM) Population Counter. This field indicates how many bits must be sampled before the measurement stops due to hitting the population limit.

Copyright © 2015 Marvell

Table 1524: EOM Set 1 Population Counter 1 Register (n=0–6)

Offset: Port0: 0x000A00AC, Port1: 0x000A08AC, Port2: 0x000A10AC, Port3: 0x000A18AC, Port4: 0x000A20AC, Port5: 0x000A28AC,

Offset Formula: 0x000A00AC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	EOM_SET1_POP_CNT[47:32]	RW 0xFFFF	Eye-opening Measurement (EOM) Population Counter. This field indicates how many bits must be sampled before the measurement stops due to hitting the population limit.

Table 1525: EOM Set 1 Population Counter 2 Register (n=0–6)

Offset: Port0: 0x000A00B0, Port1: 0x000A08B0, Port2: 0x000A10B0, Port3: 0x000A18B0, Port4: 0x000A20B0, Port5: 0x000A28B0,

Offset Formula: 0x000A00B0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	EOM_SET1_POP_CNT[31:16]	RW 0xFFFF	Eye-opening Measurement (EOM) Population Counter. This field indicates how many bits must be sampled before the measurement stops due to hitting the population limit.

Table 1526: EOM Set 1 Population Counter 3 Register (n=0–6)

Offset: Port0: 0x000A00B4, Port1: 0x000A08B4, Port2: 0x000A10B4, Port3: 0x000A18B4, Port4: 0x000A20B4, Port5: 0x000A28B4,

Offset Formula: 0x000A00B4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_POP_CNT[15:0]	RW 0xFFFF	Eye-opening Measurement (EOM) Population Counter. This field indicates how many bits must be sampled before the measurement stops due to hitting the population limit.

Table 1527: EOM Set 1 Threshold Counter 0 Register (n=0–6)

Offset: Port0: 0x000A00B8, Port1: 0x000A08B8, Port2: 0x000A10B8, Port3: 0x000A18B8, Port4: 0x000A20B8, Port5: 0x000A28B8,

Offset Formula: 0x000A00B8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1527: EOM Set 1 Threshold Counter 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A00B8, Port1: 0x000A08B8, Port2: 0x000A10B8, Port3: 0x000A18B8, Port4: 0x000A20B8, Port5: 0x000A28B8,
Offset Formula: 0x000A00B8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_THRE_CNT[63:48]	RW 0xFFFF	Eye-opening Measurement (EOM) Threshold Counter. This field can be used to set an error threshold limit. If the measured errors are this number then, if enabled, the EOM set stops. EOM measurement stops if EOM_SET1_ERROR_CNT[63:0] greater than or equal to EOM_SET1_THRE_CNT[63:0].

Table 1528: EOM Set 1 Threshold Counter 1 Register (n=0–6)
Offset: Port0: 0x000A00BC, Port1: 0x000A08BC, Port2: 0x000A10BC, Port3: 0x000A18BC, Port4: 0x000A20BC, Port5: 0x000A28BC,
Offset Formula: 0x000A00BC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	EOM_SET1_THRE_CNT[47:32]	RW 0xFFFF	Eye-opening Measurement (EOM) Threshold Counter. This field can be used to set a error threshold limit. If the measured errors are this number then, if enabled, the EOM set stops. EOM measurement stops if EOM_SET1_ERROR_CNT greater than or equal to EOM_SET1_THRE_CNT and EOM_SET1_STOP = 10b.

Table 1529: EOM Set 1 Threshold Counter 2 Register (n=0–6)
Offset: Port0: 0x000A00C0, Port1: 0x000A08C0, Port2: 0x000A10C0, Port3: 0x000A18C0, Port4: 0x000A20C0, Port5: 0x000A28C0,
Offset Formula: 0x000A00C0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_THRE_CNT[31:16]	RW 0xFFFF	Eye-opening Measurement (EOM) Threshold Counter. This field can be used to set a error threshold limit. If the measured errors are this number then, if enabled, the EOM set stops. EOM measurement stops if EOM_SET1_ERROR_CNT greater than or equal to EOM_SET1_THRE_CNT and EOM_SET1_STOP = 10b.

Copyright © 2015 Marvell

Table 1530: EOM Set 1 Threshold Counter 3 Register (n=0–6)

Offset: Port0: 0x000A00C4, Port1: 0x000A08C4, Port2: 0x000A10C4, Port3: 0x000A18C4, Port4: 0x000A20C4, Port5: 0x000A28C4,

Offset Formula: 0x000A00C4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_THRE_CNT[15:0]	RW 0xFFFF	Eye-opening Measurement (EOM) Threshold Counter. This field can be used to set a error threshold limit. If the measured errors are bigger to this number and if EOM_SET1_STOP_EN = 1xb, EOM set1 stops.

Table 1531: EOM Set 1 Sample Counter 0 Register (n=0–6)

Offset: Port0: 0x000A00C8, Port1: 0x000A08C8, Port2: 0x000A10C8, Port3: 0x000A18C8, Port4: 0x000A20C8, Port5: 0x000A28C8,

Offset Formula: 0x000A00C8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_SAMPLE_CNT[63:48]	RO 0x0	Eye-opening Measurement (EOM) Sample Counter. This counter indicates how many bits have been sampled since the beginning of the EOM Measurement.

Table 1532: EOM Set 1 Sample Counter 1 Register (n=0–6)

Offset: Port0: 0x000A00CC, Port1: 0x000A08CC, Port2: 0x000A10CC, Port3: 0x000A18CC, Port4: 0x000A20CC, Port5: 0x000A28CC,

Offset Formula: 0x000A00CC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_SAMPLE_CNT[47:32]	RO 0x0	Eye-opening Measurement (EOM) Sample Counter. This counter indicates how many bits have been sampled since the beginning of the EOM Measurement.

Table 1533: EOM Set 1 Sample Counter 2 Register (n=0–6)

Offset: Port0: 0x000A00D0, Port1: 0x000A08D0, Port2: 0x000A10D0, Port3: 0x000A18D0, Port4: 0x000A20D0, Port5: 0x000A28D0,

Offset Formula: 0x000A00D0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_SAMPLE_CNT[31:16]	RO 0x0	Eye-opening Measurement (EOM) Sample Counter. This counter indicates how many bits have been sampled since the beginning of the EOM Measurement.

Copyright © 2015 Marvell

Table 1534: EOM Set 1 Sample Counter 3 Register (n=0–6)

Offset: Port0: 0x000A00D4, Port1: 0x000A08D4, Port2: 0x000A10D4, Port3: 0x000A18D4, Port4: 0x000A20D4, Port5: 0x000A28D4,

Offset Formula: 0x000A00D4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_SAMPLE_CNT[15:0]	RO 0x0	Eye-opening Measurement (EOM) Sample Counter. This counter indicates how many bits have been sampled since the beginning of the EOM Measurement.

Table 1535: EOM Set 1 Error Counter 0 Register (n=0–6)

Offset: Port0: 0x000A00D8, Port1: 0x000A08D8, Port2: 0x000A10D8, Port3: 0x000A18D8, Port4: 0x000A20D8, Port5: 0x000A28D8,

Offset Formula: 0x000A00D8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	EOM_SET1_ERROR_CNT[63:48]	RO 0x0	Eye-opening Measurement (EOM) Error Counter. This counter indicates how many error bits have been detected since the beginning of the EOM Measurement.

Table 1536: EOM Set 1 Error Counter 1 Register (n=0–6)

Offset: Port0: 0x000A00DC, Port1: 0x000A08DC, Port2: 0x000A10DC, Port3: 0x000A18DC, Port4: 0x000A20DC, Port5: 0x000A28DC,

Offset Formula: 0x000A00DC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_ERROR_CNT[47:32]	RO 0x0	Eye-opening Measurement (EOM) Error Counter. This counter indicates how many error bits have been detected since the beginning of the EOM Measurement.

Table 1537: EOM Set 1 Error Counter 2 Register (n=0–6)

Offset: Port0: 0x000A00E0, Port1: 0x000A08E0, Port2: 0x000A10E0, Port3: 0x000A18E0, Port4: 0x000A20E0, Port5: 0x000A28E0,

Offset Formula: 0x000A00E0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	EOM_SET1_ERROR_CNT[31:16]	RO 0x0	Eye-opening Measurement (EOM) Error Counter. This counter indicates how many error bits have been detected since the beginning of the EOM Measurement.

Copyright © 2015 Marvell

Table 1538: EOM Set 1 Error Counter 3 Register (n=0–6)

Offset: Port0: 0x000A00E4, Port1: 0x000A08E4, Port2: 0x000A10E4, Port3: 0x000A18E4, Port4: 0x000A20E4, Port5: 0x000A28E4,

Offset Formula: 0x000A00E4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	EOM_SET1_ERROR_CNT[15:0]	RO 0x0	Eye-opening Measurement (EOM) Error Counter. This counter indicates how many error bits have been detected since the beginning of the EOM Measurement.

Table 1539: External Digital Reserved 0 Register (n=0–6)

Offset: Port0: 0x000A00E8, Port1: 0x000A08E8, Port2: 0x000A10E8, Port3: 0x000A18E8, Port4: 0x000A20E8, Port5: 0x000A28E8,

Offset Formula: 0x000A00E8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	DIG_RSVD0[15:0]	RW 0xF	Digital Reserved Register 0 These bits are reserved for future use by the digital design team. dig_rsvd0[15:14] is reserved for PHY software programming and thus cannot be used for PHY functional control.

Table 1540: DFE Step Size 0 Register (n=0–6)

Offset: Port0: 0x000A00EC, Port1: 0x000A08EC, Port2: 0x000A10EC, Port3: 0x000A18EC, Port4: 0x000A20EC, Port5: 0x000A28EC,

Offset Formula: 0x000A00EC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	DFE_STEP_FINE_EO_DN[3:0]	RW 0x6	dfe down step used for eye open measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Copyright © 2015 Marvell

Table 1540: DFE Step Size 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A00EC, Port1: 0x000A08EC, Port2: 0x000A10EC, Port3: 0x000A18EC, Port4: 0x000A20EC, Port5: 0x000A28EC,
Offset Formula: 0x000A00EC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11:8	DFE_STEP_FINE_EO_UP[3:0]	RW 0xA	dfe up step used for eye open measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
7:4	DFE_STEP_FINE_F0[3:0]	RW 0x6	Decision Feedback Equalization (DFE) Step Fine for F0 This field controls the fine adjust step size for the DFE training (fraction of 1 LSB per DFE update). The fine adjust is used at the end of the DFE training 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
3:0	DFE_STEP_COARSE_F0[3:0]	RW 0x2	Decision Feedback Equalization (DFE) Step coarse for F0 This field controls the fine adjust step size for the DFE training (fraction of 1 LSB per DFE update). The fine adjust is used at the end of the DFE training 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Copyright © 2015 Marvell

Table 1541: DFE Step Size 1 Register (n=0–6)

Offset: Port0: 0x000A00F0, Port1: 0x000A08F0, Port2: 0x000A10F0, Port3: 0x000A18F0, Port4: 0x000A20F0, Port5: 0x000A28F0,

Offset Formula: 0x000A00F0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	DFE_STEP_COARSE_EO_DN[3:0]	RW 0x1	dfe down step used for eye open measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
11:8	DFE_STEP_COARSE_EO_UP[3:0]	RW 0x9	dfe up step used for eye open measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
7:4	DFE_STEP_ACCURATE_F0[3:0]	RW 0x8	Decision Feedback Equalization (DFE) Step Fine for F0 This field controls the fine adjust step size for the DFE training (fraction of 1 LSB per DFE update). The fine adjust is used at the end of the DFE training 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Copyright © 2015 Marvell

Table 1541: DFE Step Size 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A00F0, Port1: 0x000A08F0, Port2: 0x000A10F0, Port3: 0x000A18F0, Port4: 0x000A20F0, Port5: 0x000A28F0,
Offset Formula: 0x000A00F0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	DFE_STEP_ACCURATE_FX[3:0]	RW 0x8	Decision Feedback Equalization (DFE) Step accurate for F0 This field controls the fine adjust step size for the DFE training (fraction of 1 LSB per DFE update). The fine adjust is used at the end of the DFE training 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Table 1542: Generation 1 Setting 2 Register (n=0–6)
Offset: Port0: 0x000A00F4, Port1: 0x000A08F4, Port2: 0x000A10F4, Port3: 0x000A18F4, Port4: 0x000A20F4, Port5: 0x000A28F4,
Offset Formula: 0x000A00F4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	G1_TX_SSC_AMP[6:0]	RW 0x18	Spread Spectrum clock Amplitude The SSC amplitude (in ppm) = 122 x field value, when PHY_MODE is set to SATA, SAS The SSC amplitude (in ppm) = 126 x field value, when PHY_MODE is set to FC The SSC amplitude (in ppm) = 129 x field value, when PHY_MODE is set to PCIE gen 1 and gen 2 and USB The SSC amplitude (in ppm) = 118 x field value, when PHY_MODE is set to PCIE gen 3
8:7	G1_VREG_RXTX_MAS_ISET[1:0]	RW 0x2	Control the master regulator unit current setting. 0 = V60u: 60u 1 = V80u: 80u 2 = V100u: 100u 3 = V120u: 120u
6:5	G1_TX_EMPH_MODE[1:0]	RW 0x1	TX emphasis mode 0 = 1/2 and 1/2 emphasis 0 = emphasis step is % of the main amplitude 1 = De-emphasis 1 = emphasis step is mV
4	G1_TX_EMPH0_EN	RW 0x0	Transmitter Pre Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 0

Table 1542: Generation 1 Setting 2 Register (n=0–6) (Continued)
Offset: Port0: 0x000A00F4, Port1: 0x000A08F4, Port2: 0x000A10F4, Port3: 0x000A18F4, Port4: 0x000A20F4, Port5: 0x000A28F4,
Offset Formula: 0x000A00F4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	G1_TX_EMPH0[3:0]	RW 0x0	Transmitter Pre Emphasis Value This setting is used for PIN_PHY_GEN_TX is 0

Table 1543: Generation 2 Setting 2 Register (n=0–6)
Offset: Port0: 0x000A00F8, Port1: 0x000A08F8, Port2: 0x000A10F8, Port3: 0x000A18F8, Port4: 0x000A20F8, Port5: 0x000A28F8,
Offset Formula: 0x000A00F8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	G2_TX_SSC_AMP[6:0]	RW 0x18	Spread Spectrum clock Amplitude The SSC amplitude (in ppm) = 122 x field value, when PHY_MODE is set to SATA, SAS The SSC amplitude (in ppm) = 126 x field value, when PHY_MODE is set to FC The SSC amplitude (in ppm) = 129 x field value, when PHY_MODE is set to PCIE gen 1 and gen 2 and USB The SSC amplitude (in ppm) = 118 x field value, when PHY_MODE is set to PCIE gen 3
8:7	G2_VREG_RXTX_MAS_ISET[1:0]	RW 0x2	Control the master regulator unit current setting. 0 = V60u: 60u 1 = V80u: 80u 2 = V100u: 100u 3 = V120u: 120u
6:5	G2_TX_EMPH_MODE[1:0]	RW 0x1	TX emphasis mode 0 = 1/2 and 1/2 emphasis 0 = emphasis step is % of the main amplitude 1 = De-emphasis 1 = emphasis step is mV
4	G2_TX_EMPH0_EN	RW 0x0	Transmitter Pre Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 1
3:0	G2_TX_EMPH0[3:0]	RW 0x0	Transmitter Pre Emphasis Value This setting is used for PIN_PHY_GEN_TX is 1

Copyright © 2015 Marvell

Table 1544: Generation 3 Setting 2 Register (n=0–6)

Offset: Port0: 0x000A00FC, Port1: 0x000A08FC, Port2: 0x000A10FC, Port3: 0x000A18FC, Port4: 0x000A20FC, Port5: 0x000A28FC,

Offset Formula: 0x000A00FC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	G3_TX_SSC_AMP[6:0]	RW 0x18	Spread Spectrum clock Amplitude The SSC amplitude (in ppm) = 122 x field value, when PHY_MODE is set to SATA, SAS The SSC amplitude (in ppm) = 126 x field value, when PHY_MODE is set to FC The SSC amplitude (in ppm) = 129 x field value, when PHY_MODE is set to PCIE gen 1 and gen 2 and USB The SSC amplitude (in ppm) = 118 x field value, when PHY_MODE is set to PCIE gen 3
8:7	G3_VREG_RXTX_MAS_ISET[1:0]	RW 0x2	Control the master regulator unit current setting. 0 = V60u: 60u 1 = V80u: 80u 2 = V100u: 100u 3 = V120u: 120u
6:5	G3_TX_EMPH_MODE[1:0]	RW 0x2	TX emphasis mode 0 = 1/2 and 1/2 emphasis 0 = emphasis step is % of the main amplitude 1 = De-emphasis 1 = emphasis step is mV
4	G3_TX_EMPH0_EN	RW 0x0	Transmitter Pre Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 2
3:0	G3_TX_EMPH0[3:0]	RW 0x0	Transmitter Pre Emphasis Value This setting is used for PIN_PHY_GEN_TX is 2

Table 1545: Generation 4 Setting 2 Register (n=0–6)

Offset: Port0: 0x000A0100, Port1: 0x000A0900, Port2: 0x000A1100, Port3: 0x000A1900, Port4: 0x000A2100, Port5: 0x000A2900,

Offset Formula: 0x000A0100+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	G4_TX_SSC_AMP[6:0]	RW 0x18	Spread Spectrum clock Amplitude The SSC amplitude (in ppm) = 122 x field value, when PHY_MODE is set to SATA, SAS The SSC amplitude (in ppm) = 126 x field value, when PHY_MODE is set to FC The SSC amplitude (in ppm) = 129 x field value, when PHY_MODE is set to PCIe gen 1 and gen 2 and USB The SSC amplitude (in ppm) = 118 x field value, when PHY_MODE is set to PCIe gen 3

Copyright © 2015 Marvell

Table 1545: Generation 4 Setting 2 Register (n=0–6) (Continued)

Offset: Port0: 0x000A0100, Port1: 0x000A0900, Port2: 0x000A1100, Port3: 0x000A1900, Port4: 0x000A2100, Port5: 0x000A2900,

Offset Formula: 0x000A0100+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
8:7	G4_VREG_RXTX_MAS_ISET[1:0]	RW 0x2	Control the master regulator unit current setting. 0 = V60u: 60u 1 = V80u: 80u 2 = V100u: 100u 3 = V120u: 120u
6:5	G4_TX_EMPH_MODE[1:0]	RW 0x2	TX emphasis mode 0 = 1/2 and 1/2 emphasis 0 = emphasis step is % of the main amplitude 1 = De-emphasis 1 = emphasis step is mV
4	G4_TX_EMPH0_EN	RW 0x0	Transmitter Pre Emphasis Enable This setting is used for PIN_PHY_GEN_TX is 3
3:0	G4_TX_EMPH0[3:0]	RW 0x0	Transmitter Pre Emphasis Value This setting is used for PIN_PHY_GEN_TX is 3

Table 1546: VTHIMPCAL Value Control Register (n=0–6)

Offset: Port0: 0x000A0104, Port1: 0x000A0904, Port2: 0x000A1104, Port3: 0x000A1904, Port4: 0x000A2104, Port5: 0x000A2904,

Offset Formula: 0x000A0104+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	Reserved	RSVD 0xA	Reserved
11:8	RXIMPCAL_TH[3:0]	RW 0xA	RX Impedance calibration threshold voltage setting. . . . . . . 0 = V40ohm: 40ohm 1 = 41.25ohm 8 = V50ohm: 50ohm 15 = 58.75ohm
7:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3:0	Reserved	RSVD 0x8	Reserved

Copyright © 2015 Marvell

Table 1547: VDD Calibration Control 0 Register (n=0–6)

Offset: Port0: 0x000A0108, Port1: 0x000A0908, Port2: 0x000A1108, Port3: 0x000A1908, Port4: 0x000A2108, Port5: 0x000A2908,

Offset Formula: 0x000A0108+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:5	Reserved	RSVD 0x121	Reserved
4:0	EXT_SELLV_RXDIG[4:0]	RW 0x10	External RXDIG Regulator Control Value

Table 1548: VDD Calibration Control 1 Register (n=0–6)

Offset: Port0: 0x000A010C, Port1: 0x000A090C, Port2: 0x000A110C, Port3: 0x000A190C, Port4: 0x000A210C, Port5: 0x000A290C,

Offset Formula: 0x000A010C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:11	Reserved	RSVD 0x1	Reserved
10:9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8:0	Reserved	RSVD 0x30	Reserved

Table 1549: VDD Calibration Control 2 Register (n=0–6)

Offset: Port0: 0x000A0110, Port1: 0x000A0910, Port2: 0x000A1110, Port3: 0x000A1910, Port4: 0x000A2110, Port5: 0x000A2910,

Offset Formula: 0x000A0110+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1030	Reserved

Table 1550: VDD Calibration Control 3 Register (n=0–6)

Offset: Port0: 0x000A0114, Port1: 0x000A0914, Port2: 0x000A1114, Port3: 0x000A1914, Port4: 0x000A2114, Port5: 0x000A2914,

Offset Formula: 0x000A0114+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	Reserved	RSVD 0x11	Reserved
9:5	EXT_SELLV_RXSAMPL[4:0]	RW 0x10	External RX Regulator Control Value
4:0	Reserved	RSVD 0x10	Reserved

Copyright © 2015 Marvell

Table 1551: Analog Reserved 1 Register (n=0–6)

Offset: Port0: 0x000A0118, Port1: 0x000A0918, Port2: 0x000A1118, Port3: 0x000A1918, Port4: 0x000A2118, Port5: 0x000A2918,

Offset Formula: 0x000A0118+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	ANA_RSVD0[15:0]	RW 0xFF00	Analog Reserved 0 These bits are reserved for future use by the analog design team.

Table 1552: Analog Reserved 2 Register (n=0–6)

Offset: Port0: 0x000A011C, Port1: 0x000A091C, Port2: 0x000A111C, Port3: 0x000A191C, Port4: 0x000A211C, Port5: 0x000A291C,

Offset Formula: 0x000A011C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	ANA_RSVD1[15:0]	RW 0xFF00	Analog Reserved 1 These bits are reserved for future use by the analog design team.

Table 1553: PCIe/USB3 Control Register (n=0–6)

Offset: Port0: 0x000A0120, Port1: 0x000A0920, Port2: 0x000A1120, Port3: 0x000A1920, Port4: 0x000A2120, Port5: 0x000A2920,

Offset Formula: 0x000A0120+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x5060	Reserved

Table 1554: Lane Alignment Control Register (n=0–6)

Offset: Port0: 0x000A0124, Port1: 0x000A0924, Port2: 0x000A1124, Port3: 0x000A1924, Port4: 0x000A2124, Port5: 0x000A2924,

Offset Formula: 0x000A0124+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:13	Reserved	RSVD 0x0	Reserved
12	LANE_ALIGN_OFF	RW 0x1	Lane Align Disable
11:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1555: Analog Test Point Selection Register (n=0–6)

Offset: Port0: 0x000A0128, Port1: 0x000A0928, Port2: 0x000A1128, Port3: 0x000A1928, Port4: 0x000A2128, Port5: 0x000A2928,

Offset Formula: 0x000A0128+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x400	Reserved

Table 1556: 150M Clock Control 0 Register (n=0–6)

Offset: Port0: 0x000A012C, Port1: 0x000A092C, Port2: 0x000A112C, Port3: 0x000A192C, Port4: 0x000A212C, Port5: 0x000A292C,

Offset Formula: 0x000A012C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x0	Reserved
14:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x40	Reserved

Table 1557: 150M Clock Control 1 Register (n=0–6)

Offset: Port0: 0x000A0130, Port1: 0x000A0930, Port2: 0x000A1130, Port3: 0x000A1930, Port4: 0x000A2130, Port5: 0x000A2930,

Offset Formula: 0x000A0130+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0xF0	Reserved

Table 1558: Regulator Output Control 0 Register (n=0–6)

Offset: Port0: 0x000A0134, Port1: 0x000A0934, Port2: 0x000A1134, Port3: 0x000A1934, Port4: 0x000A2134, Port5: 0x000A2934,

Offset Formula: 0x000A0134+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x0	Reserved
14:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:10	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1558: Regulator Output Control 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0134, Port1: 0x000A0934, Port2: 0x000A1134, Port3: 0x000A1934, Port4: 0x000A2134, Port5: 0x000A2934,
Offset Formula: 0x000A0134+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8:0	Reserved	RSVD 0x52	Reserved

Table 1559: Regulator Output Control 1 Register (n=0–6)
Offset: Port0: 0x000A0138, Port1: 0x000A0938, Port2: 0x000A1138, Port3: 0x000A1938, Port4: 0x000A2138, Port5: 0x000A2938,
Offset Formula: 0x000A0138+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:12	Reserved	RSVD 0x1	Reserved
11:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:8	Reserved	RSVD 0x1	Reserved
7:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1560: Miscellaneous Control 0 Register (n=0–6)
Offset: Port0: 0x000A013C, Port1: 0x000A093C, Port2: 0x000A113C, Port3: 0x000A193C, Port4: 0x000A213C, Port5: 0x000A293C,
Offset Formula: 0x000A013C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:15	Reserved	RSVD 0x1	Reserved
14	RX_INIT_SPD_CHG_DIS	RW 0x1	disable rx_init when speed change
13:11	Reserved	RSVD 0x4	Reserved
10	REFCLK_SEL	RW 0x0	Reference Clock Selection 0 = reference clock comes from group 1 REFCLKC_IN_RXSIDE/REFCLKC_IN_TXSIDE 1 = reference clock comes from group 2 REFCLKC_IN_RXSIDE_G2/REFCLKC_IN_TXSIDE_G2

Copyright © 2015 Marvell

Table 1560: Miscellaneous Control 0 Register (n=0–6) (Continued)
Offset: Port0: 0x000A013C, Port1: 0x000A093C, Port2: 0x000A113C, Port3: 0x000A193C, Port4: 0x000A213C, Port5: 0x000A293C,
Offset Formula: 0x000A013C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
9	Reserved	RSVD 0x0	Reserved
8	CLK100M_125M_SEL	RW 0x0	PIN_CLK100M_125M Clock Frequency Selection 0 = PIN_CLK100M_125M sends out 100MHz clock 1 = PIN_CLK100M_125M sends out 125MHz clock
7	CLK500M_EN	RW 0x1	500MHz Clock Enable 0 = Disable 500MHz clock 1 = Enable 500MHz clock
6	TXDCLK_2X_SEL	RW 0x1	PIN_TXDCLK_2X Clock Frequency Selection 0 = PIN_TXDCLK_2X sends out TXCLK_2X 1 = PIN_TXDCLK_2X sends out 500MHz clock
5	ICP_FORCE	RW 0x0	Forcing the ICP[4:0] value to come from register field instead of using the value from the automated lookup table.
4	CLK100M_125M_EN	RW 0x0	PIN_CLK100M_125M enable
3:0	ICP[3:0]	RW 0xA	PLL Charge-pump Current Control. If ICP_FORCE = 1, then this field controls the ICP signal instead of the automatic lookup table. Charge pump current setting 0 = 37.5uA 1 = V45uA: 45uA 2 = 52.5uA 3 = V60uA: 60uA 4 = V75uA: 75uA 5 = V90uA: 90uA 6 = V105uA: 105uA 7 = V120uA: 120uA 8 = V150uA: 150uA 9 = V180uA: 180uA 10 = V210uA: 210uA 11 = V240uA: 240uA 12 = V300uA: 300uA 13 = V360uA: 360uA 14 = V420uA: 420uA 15 = V480uA: 480uA

Copyright © 2015 Marvell

Table 1561: RX Control 1 Register (n=0–6)

Offset: Port0: 0x000A0140, Port1: 0x000A0940, Port2: 0x000A1140, Port3: 0x000A1940, Port4: 0x000A2140, Port5: 0x000A2940,

Offset Formula: 0x000A0140+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:13	Reserved	RSVD 0x2	Reserved
12	CLK8T_EN	RW 0x1	Enable RX and TX 8T Clock. This bit will not affect PIN_TXDCLK_2X. When this bit is high, PIN_TXDCLK_8T will output 8T clock This bit controls PIN_RXDCLK_2X 0 = determined by rxclk2x_sel 0 = bit rate / 20 (40-bit mode), bit rate / 16 (32-bit mode) 1 = bit rate / 8 1 = bit rate / 8
11	RXCLK2X_SEL	RW 0x1	PIN_RXDCLK_2X Source Selection in 20/16-bit Mode in 20/16-bit mode 0 = bit rate / 20 (20-bit mode), bit rate / 16 (16-bit mode) 1 = bit rate / 10 (20-bit mode), bit rate / 8 (16-bit mode)
10:0	Reserved	RSVD 0x0	Reserved

Table 1562: Interface and Digital Reference Clock Control Register (n=0–6)

Offset: Port0: 0x000A0144, Port1: 0x000A0944, Port2: 0x000A1144, Port3: 0x000A1944, Port4: 0x000A2144, Port5: 0x000A2944,

Offset Formula: 0x000A0144+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:9	Reserved	RSVD 0x1	Reserved
8	REF1M_GEN_DIV_FORCE	RW 0x0	Reference 1 MHz Generation Divider Force.
7:0	REF1M_GEN_DIV[7:0]	RW 0x80	Reference 1 MHz Generation Divider.

Copyright © 2015 Marvell

Table 1563: Power Control Register (n=0–6)

Offset: Port0: 0x000A0148, Port1: 0x000A0948, Port2: 0x000A1148, Port3: 0x000A1948, Port4: 0x000A2148, Port5: 0x000A2948,

Offset Formula: 0x000A0148+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:11	Reserved	RSVD 0x1C	Reserved
10	SFT_RST_NO_REG	RW 0x0	Software Reset for Internal Logic. soft reset except register, need write 0 to release from reset
9	SFT_RST	RW 0x0	Soft Reset for Registers Resets all registers to the default value. This control only resets the registers and doesn't reset any state machines. Or with chip_reset to generate reset for registers.
8:0	Reserved	RSVD 0x8	Reserved

Table 1564: Feedback Divider Register (n=0–6)

Offset: Port0: 0x000A014C, Port1: 0x000A094C, Port2: 0x000A114C, Port3: 0x000A194C, Port4: 0x000A214C, Port5: 0x000A294C,

Offset Formula: 0x000A014C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x84B4	Reserved

Table 1565: Forward Divider Register (n=0–6)

Offset: Port0: 0x000A0150, Port1: 0x000A0950, Port2: 0x000A1150, Port3: 0x000A1950, Port4: 0x000A2150, Port5: 0x000A2950,

Offset Formula: 0x000A0150+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x5001	Reserved

Table 1566: Transmitter/Receiver Speed Divider Force Register (n=0–6)

Offset: Port0: 0x000A0154, Port1: 0x000A0954, Port2: 0x000A1154, Port3: 0x000A1954, Port4: 0x000A2154, Port5: 0x000A2954,

Offset Formula: 0x000A0154+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:7	Reserved	RSVD 0x0	Reserved
6:5	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1566: Transmitter/Receiver Speed Divider Force Register (n=0–6) (Continued)
Offset: Port0: 0x000A0154, Port1: 0x000A0954, Port2: 0x000A1154, Port3: 0x000A1954, Port4: 0x000A2154, Port5: 0x000A2954,
Offset Formula: 0x000A0154+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
4:0	Reserved	RSVD 0x7	Reserved

Table 1567: PLL Bandwidth Adjust Register (n=0–6)
Offset: Port0: 0x000A0158, Port1: 0x000A0958, Port2: 0x000A1158, Port3: 0x000A1958, Port4: 0x000A2158, Port5: 0x000A2958,
Offset Formula: 0x000A0158+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	SPEED_CHANGE_CLK_ON	RW 0x0	Speed Change Clock On 0 = TXDCLK or RXDCLK is off for 2 cycles during TX or RX speed change 1 = TXDCLK and RXDCLK are on during TX or RX speed change
14:0	Reserved	RSVD 0x4000	Reserved

Table 1568: Calibration Enable Control Register (n=0–6)
Offset: Port0: 0x000A015C, Port1: 0x000A095C, Port2: 0x000A115C, Port3: 0x000A195C, Port4: 0x000A215C, Port5: 0x000A295C,
Offset Formula: 0x000A015C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	TXDCC_FB_POINT_SEL	RW 0x1	

Copyright © 2015 Marvell

Table 1569: TX DCC Control Register (n=0–6)

Offset: Port0: 0x000A0160, Port1: 0x000A0960, Port2: 0x000A1160, Port3: 0x000A1960, Port4: 0x000A2160, Port5: 0x000A2960,

Offset Formula: 0x000A0160+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	TXDCC_CAL_STOP_SEL	RW 0x0	Transmitter DCC Calibration Stop Select. When txdcc_cal_stop_sel = 1, DCC calibration for transmitter only runs for limited time. The duration of this calibration is decided by txdcc_countsize. When txdcc_cal_stop_sel = 0, calibration runs from cal_txdcc_start to txdcc_cal_freeze txdcc_countsize: Duration (us) when 3'b000: 15360 us when 3'b001: 7680 us when 3'b010: 3840 us when 3'b011: 1920 us when 3'b100: 960 us when 3'b101: 480us Other: reserved 0 = DCC calibration/adaptation continues unless txdcc_cal_freeze = 1. 1 = Calibration/adaptation stops after a number of iterations programmable by countsize txdcc_countsize.
14:0	Reserved	RSVD 0x20A	Reserved

Table 1570: RX DCC Edge and Center Control Register (n=0–6)

Offset: Port0: 0x000A0164, Port1: 0x000A0964, Port2: 0x000A1164, Port3: 0x000A1964, Port4: 0x000A2164, Port5: 0x000A2964,

Offset Formula: 0x000A0164+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x8820	Reserved

Table 1571: RX CLK Align 90 and TX Idle Calibration Control Register (n=0–6)

Offset: Port0: 0x000A0168, Port1: 0x000A0968, Port2: 0x000A1168, Port3: 0x000A1968, Port4: 0x000A2168, Port5: 0x000A2968,

Offset Formula: 0x000A0168+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:6	Reserved	RSVD 0x180	Reserved

Copyright © 2015 Marvell

Table 1571: RX CLK Align 90 and TX Idle Calibration Control Register (n=0–6) (Continued)
Offset: Port0: 0x000A0168, Port1: 0x000A0968, Port2: 0x000A1168, Port3: 0x000A1968, Port4: 0x000A2168, Port5: 0x000A2968,
Offset Formula: 0x000A0168+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
5	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
4:0	Reserved	RSVD 0x14	Reserved

Table 1572: Sampler and Process Calibration Control Register (n=0–6)
Offset: Port0: 0x000A016C, Port1: 0x000A096C, Port2: 0x000A116C, Port3: 0x000A196C, Port4: 0x000A216C, Port5: 0x000A296C,
Offset Formula: 0x000A016C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x8054	Reserved

Table 1573: Process Threshold Setting Register (n=0–6)
Offset: Port0: 0x000A0170, Port1: 0x000A0970, Port2: 0x000A1170, Port3: 0x000A1970, Port4: 0x000A2170, Port5: 0x000A2970,
Offset Formula: 0x000A0170+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x1	Reserved
14:10	PROCESS_CAL_SUBSS[4:0]	RW 0xA	Process Calibration Threshold for the Initial Counter.
9:5	PROCESS_CAL_SS2TT[4:0]	RW 0x8	Process Calibration Threshold for the Slow to Faster Corner Counter.
4:0	PROCESS_CAL_TT2FF[4:0]	RW 0x15	Process Calibration Threshold for the Typical to Faster Corner Counter.

Table 1574: Transmitter Slew Rate Control Register (n=0–6)
Offset: Port0: 0x000A0174, Port1: 0x000A0974, Port2: 0x000A1174, Port3: 0x000A1974, Port4: 0x000A2174, Port5: 0x000A2974,
Offset Formula: 0x000A0174+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0xFCC8	Reserved

Copyright © 2015 Marvell

Table 1575: Transmitter Amplitude and Emphasis Control Register (n=0–6)

Offset: Port0: 0x000A0178, Port1: 0x000A0978, Port2: 0x000A1178, Port3: 0x000A1978, Port4: 0x000A2178, Port5: 0x000A2978,

Offset Formula: 0x000A0178+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:0	Reserved	RSVD 0x2A00	Reserved

Table 1576: SSC Control Register (n=0–6)

Offset: Port0: 0x000A017C, Port1: 0x000A097C, Port2: 0x000A117C, Port3: 0x000A197C, Port4: 0x000A217C, Port5: 0x000A297C,

Offset Formula: 0x000A017C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x0	Reserved
14	SSC_EN_FROM_PIN	RW 0x1	SSC Enable Control from PIN 0 = Select ssc_en from register or G-table 1 = Select ssc_en from input PIN
13:0	Reserved	RSVD 0x5CF	Reserved

Table 1577: SSC Step Register (n=0–6)

Offset: Port0: 0x000A0180, Port1: 0x000A0980, Port2: 0x000A1180, Port3: 0x000A1980, Port4: 0x000A2180, Port5: 0x000A2980,

Offset Formula: 0x000A0180+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:0	Reserved	RSVD 0xF	Reserved

Copyright © 2015 Marvell

Table 1578: DTL Control Register (n=0–6)

Offset: Port0: 0x000A0184, Port1: 0x000A0984, Port2: 0x000A1184, Port3: 0x000A1984, Port4: 0x000A2184, Port5: 0x000A2984,

Offset Formula: 0x000A0184+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	Reserved	RSVD 0x1	Reserved
11:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:3	Reserved	RSVD 0x2	Reserved
2	DTL_FLOOP_EN	RW 0x1	DTL Frequency Loop Enable. This configures the CDR to turn on the frequency loop. This is required to support the reception of SSC, or to handle large frequency offsets between the incoming data and the local clock. 0 = frequency loop set to initial value 1 = frequency loop is enabled
1:0	Reserved	RSVD 0x3	Reserved

Table 1579: Offset Phase Control Register (n=0–6)

Offset: Port0: 0x000A0188, Port1: 0x000A0988, Port2: 0x000A1188, Port3: 0x000A1988, Port4: 0x000A2188, Port5: 0x000A2988,

Offset Formula: 0x000A0188+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x2860	Reserved

Table 1580: Receiver Frequency Offset Extraction Register (n=0–6)

Offset: Port0: 0x000A018C, Port1: 0x000A098C, Port2: 0x000A118C, Port3: 0x000A198C, Port4: 0x000A218C, Port5: 0x000A298C,

Offset Formula: 0x000A018C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x64F	Reserved

Table 1581: Receiver Frequency Offset Read Register (n=0–6)

Offset: Port0: 0x000A0190, Port1: 0x000A0990, Port2: 0x000A1190, Port3: 0x000A1990, Port4: 0x000A2190, Port5: 0x000A2990,

Offset Formula: 0x000A0190+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1581: Receiver Frequency Offset Read Register (n=0–6) (Continued)
Offset: Port0: 0x000A0190, Port1: 0x000A0990, Port2: 0x000A1190, Port3: 0x000A1990, Port4: 0x000A2190, Port5: 0x000A2990,
Offset Formula: 0x000A0190+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
14:0	Reserved	RSVD 0x0	Reserved

Table 1582: Receiver Offset Control Register (n=0–6)
Offset: Port0: 0x000A0194, Port1: 0x000A0994, Port2: 0x000A1194, Port3: 0x000A1994, Port4: 0x000A2194, Port5: 0x000A2994,
Offset Formula: 0x000A0194+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x204	Reserved

Table 1583: Receiver Clamping Control Register (n=0–6)
Offset: Port0: 0x000A0198, Port1: 0x000A0998, Port2: 0x000A1198, Port3: 0x000A1998, Port4: 0x000A2198, Port5: 0x000A2998,
Offset Formula: 0x000A0198+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x800	Reserved

Table 1584: DFE Miscellaneous Control Register (n=0–6)
Offset: Port0: 0x000A019C, Port1: 0x000A099C, Port2: 0x000A119C, Port3: 0x000A199C, Port4: 0x000A219C, Port5: 0x000A299C,
Offset Formula: 0x000A019C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	DFE_INIT_DIS	RW 0x0	DFE Initialization Disable It is used to control if DFE adaptation initial value should be reloaded or kept unchanged during RX initialization. By setting this bit to 1, DFE adapted value can be kept during mode changes, such as partial or slumber mode to normal mode 0 = enable DFE loading during rx_init 1 = disable DFE loading during rx_init
14:12	Reserved	RSVD 0x6	Reserved
11	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
10:1	Reserved	RSVD 0x4	Reserved
0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1585: Data Phase Offset Control Register (n=0–6)

Offset: Port0: 0x000A01A0, Port1: 0x000A09A0, Port2: 0x000A11A0, Port3: 0x000A19A0, Port4: 0x000A21A0, Port5: 0x000A29A0,

Offset Formula: 0x000A01A0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x10F	Reserved

Table 1586: Adapted FFE Capacitor/Counter Control Register (n=0–6)

Offset: Port0: 0x000A01A4, Port1: 0x000A09A4, Port2: 0x000A11A4, Port3: 0x000A19A4, Port4: 0x000A21A4, Port5: 0x000A29A4,

Offset Formula: 0x000A01A4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x0	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:4	Reserved	RSVD 0x2	Reserved
3	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
2:0	Reserved	RSVD 0x3	Reserved

Table 1587: Error Counter Constraint Control Register (n=0–6)

Offset: Port0: 0x000A01A8, Port1: 0x000A09A8, Port2: 0x000A11A8, Port3: 0x000A19A8, Port4: 0x000A21A8, Port5: 0x000A29A8,

Offset Formula: 0x000A01A8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14	Reserved	RSVD 0x1	Reserved
13	FAST_DFE_TIMER_EN	RW 0x0	Fast DFE Timer Enable If this bit is high, timer used in start up state machine is 1/4 of normal one
12:0	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1588: DFE Timer Register (n=0–6)

Offset: Port0: 0x000A01AC, Port1: 0x000A09AC, Port2: 0x000A11AC, Port3: 0x000A19AC, Port4: 0x000A21AC, Port5: 0x000A29AC,

Offset Formula: 0x000A01AC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	DFE_INIT_REG	RW 0x0	Decision Feedback Equalization (DFE) Initial Register. This control selects which DFE tap values are used as a starting point after PIN_RX_INIT is pulsed. Pulsing PIN_RX_INIT starts the initialization state machines for the receiver that locks the CDR, and trains the DFE. 0 = DFE uses the F0-F5 values from the previous training result 1 = DFE uses the default values for F0-F5
14	RX_FAST_ACQ_EN	RW 0x1	Receiver Fast Acquisition Enable.
13:12	RX_FAST_ACQ_SEL[1:0]	RW 0x0	Fast Acquisition Select. Selects the fast acquisition timer length. 0 = x1 1 = x2 2 = x4 3 = x8
11:9	TIMER_ACQ[2:0]	RW 0x2	Timer Acquisition. This field controls the amount of time in coarse adjust phase of the CDR acquisition. 0 = 20x2^10 UI 1 = 20x2^11 UI 2 = 20x2^12 UI 3 = 20x2^13 UI 4 = 20x2^14 UI 5 = 20x2^15 UI 6 = 20x2^16 UI 7 = 20x2^17 UI
8:6	Reserved	RSVD 0x3	Reserved
5:3	TIMER_DFE2[2:0]	RW 0x3	Timer Decision Feedback Equalization (DFE) 2. This field controls the amount of time spent in fine adjust phase of both DFE and CDR adaptation while in the acquisition state ST_DTL_ACQ or ST_STARTUP before PHY_RDY is asserted. 0 = 20x2^10 UI 1 = 20x2^11 UI 2 = 20x2^12 UI 3 = 20x2^13 UI 4 = 20x2^14 UI 5 = 20x2^15 UI 6 = 20x2^16 UI 7 = 20x2^17 UI
2:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1589: Eye Threshold Mode Selection Register (n=0–6)

Offset: Port0: 0x000A01B0, Port1: 0x000A09B0, Port2: 0x000A11B0, Port3: 0x000A19B0, Port4: 0x000A21B0, Port5: 0x000A29B0,

Offset Formula: 0x000A01B0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x214	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1590: RX SSC Extraction Register (n=0–6)

Offset: Port0: 0x000A01B4, Port1: 0x000A09B4, Port2: 0x000A11B4, Port3: 0x000A19B4, Port4: 0x000A21B4, Port5: 0x000A29B4,

Offset Formula: 0x000A01B4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:0	Reserved	RSVD 0x0	Reserved

Table 1591: Digital Test Bus Configuration Register (n=0–6)

Offset: Port0: 0x000A01B8, Port1: 0x000A09B8, Port2: 0x000A11B8, Port3: 0x000A19B8, Port4: 0x000A21B8, Port5: 0x000A29B8,

Offset Formula: 0x000A01B8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3	SQ_THRESH_IN_CHANGE_CAL_EN	RW 0x0	Enables the feature that the change of reg_sq_thresh_in[3:0] triggers squelch calibration 0 = the change of reg_sq_thresh_in[3:0] does not trigger squelch calibration 1 = the change of reg_sq_thresh_in[3:0] triggers squelch calibration
2	Reserved	RSVD 0x1	Reserved

Copyright © 2015 Marvell

Table 1591: Digital Test Bus Configuration Register (n=0–6) (Continued)
Offset: Port0: 0x000A01B8, Port1: 0x000A09B8, Port2: 0x000A11B8, Port3: 0x000A19B8, Port4: 0x000A21B8, Port5: 0x000A29B8,
Offset Formula: 0x000A01B8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
1:0	SQ_SEL_IN[1:0]	RW 0x2	Squelch Calibration Mode 2 Peak Detector Input Polarity Setting 0 = Calibrate mode 2 once with peak detector input set to positive 1 = Calibrate mode 2 twice with peak detector input set to positive, then negative; output the comparator calibration result with the smallest absolute value 2 = Calibrate mode 2 twice with peak detector input set to positive, then negative; output the averaged comparator calibration result 3 = Calibrate mode 2 twice with peak detector input set to positive, then negative; output the comparator calibration result with the largest absolute value

Table 1592: External Squelch Calibration Control Register (n=0–6)
Offset: Port0: 0x000A01BC, Port1: 0x000A09BC, Port2: 0x000A11BC, Port3: 0x000A19BC, Port4: 0x000A21BC, Port5: 0x000A29BC,
Offset Formula: 0x000A01BC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0x0	Reserved
3:2	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
1:0	Reserved	RSVD 0x0	Reserved

Table 1593: PLL Calibration Control 0 Register (n=0–6)
Offset: Port0: 0x000A01C0, Port1: 0x000A09C0, Port2: 0x000A11C0, Port3: 0x000A19C0, Port4: 0x000A21C0, Port5: 0x000A29C0,
Offset Formula: 0x000A01C0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1D00	Reserved

Table 1594: PLL Calibration Control 1 Register (n=0–6)
Offset: Port0: 0x000A01C4, Port1: 0x000A09C4, Port2: 0x000A11C4, Port3: 0x000A19C4, Port4: 0x000A21C4, Port5: 0x000A29C4,
Offset Formula: 0x000A01C4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x3	Reserved

Copyright © 2015 Marvell

Table 1594: PLL Calibration Control 1 Register (n=0–6) (Continued)

Offset: Port0: 0x000A01C4, Port1: 0x000A09C4, Port2: 0x000A11C4, Port3: 0x000A19C4, Port4: 0x000A21C4, Port5: 0x000A29C4,

Offset Formula: 0x000A01C4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11:6	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
5:0	SPEED_PLL[5:0]	RO 0x0	Speed_pll auto-calibration result.

Table 1595: Squelch Glitch Filter Control Register (n=0–6)

Offset: Port0: 0x000A01C8, Port1: 0x000A09C8, Port2: 0x000A11C8, Port3: 0x000A19C8, Port4: 0x000A21C8, Port5: 0x000A29C8,

Offset Formula: 0x000A01C8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14	Reserved	RSVD 0x1	Reserved
13	SQ_GATE_RXDATA_EN	RW 0x0	Use filtered SQ output to gate PIN_RXDATA_LSB/MSB. 0 = do not use sq_out_filtered to gate PIN_RXDATA_LSB/MSB 1 = use sq_out_filtered to gate PIN_RXDATA_LSB/MSB
12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	Reserved	RSVD 0x0	Reserved
9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8:0	Reserved	RSVD 0x66	Reserved

Table 1596: Miscellaneous Control 1 Register (n=0–6)

Offset: Port0: 0x000A01CC, Port1: 0x000A09CC, Port2: 0x000A11CC, Port3: 0x000A19CC, Port4: 0x000A21CC, Port5: 0x000A29CC,

Offset Formula: 0x000A01CC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:15	Reserved	RSVD 0x0	Reserved
14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1596: Miscellaneous Control 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A01CC, Port1: 0x000A09CC, Port2: 0x000A11CC, Port3: 0x000A19CC, Port4: 0x000A21CC, Port5: 0x000A29CC,
Offset Formula: 0x000A01CC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
13:1	Reserved	RSVD 0x248	Reserved
0	BG_RDY	RW 0x1	

Table 1597: Miscellaneous Control 1 Register (n=0–6)
Offset: Port0: 0x000A01D0, Port1: 0x000A09D0, Port2: 0x000A11D0, Port3: 0x000A19D0, Port4: 0x000A21D0, Port5: 0x000A29D0,
Offset Formula: 0x000A01D0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:9	Reserved	RSVD 0x0	Reserved
8:5	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
4	Reserved	RSVD 0x0	Reserved
3	BG_RDY_FORCE	RW 0x0	Force bg_rdy signal to use register value 0 = use PIN_BG_RDY 1 = use register bg_rdy
2	Reserved	RSVD 0x0	Reserved
1	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
0	Reserved	RSVD 0x1	Reserved

Table 1598: Squelch Calibration Register (n=0–6)
Offset: Port0: 0x000A01D4, Port1: 0x000A09D4, Port2: 0x000A11D4, Port3: 0x000A19D4, Port4: 0x000A21D4, Port5: 0x000A29D4,
Offset Formula: 0x000A01D4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	Reserved	RSVD 0x3	Reserved

Copyright © 2015 Marvell

Table 1598: Squelch Calibration Register (n=0–6) (Continued)

Offset: Port0: 0x000A01D4, Port1: 0x000A09D4, Port2: 0x000A11D4, Port3: 0x000A19D4, Port4: 0x000A21D4, Port5: 0x000A29D4,

Offset Formula: 0x000A01D4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
13:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:7	Reserved	RSVD 0x15	Reserved
6	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
5:0	Reserved	RSVD 0x4	Reserved

Table 1599: Reg0p9 Control Register (n=0–6)

Offset: Port0: 0x000A01D8, Port1: 0x000A09D8, Port2: 0x000A11D8, Port3: 0x000A19D8, Port4: 0x000A21D8, Port5: 0x000A29D8,

Offset Formula: 0x000A01D8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:7	Reserved	RSVD 0x6	Reserved
6:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3	Reserved	RSVD 0x0	Reserved
2:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1600: Clk150M Control 2 Register (n=0–6)

Offset: Port0: 0x000A01DC, Port1: 0x000A09DC, Port2: 0x000A11DC, Port3: 0x000A19DC, Port4: 0x000A21DC, Port5: 0x000A29DC,

Offset Formula: 0x000A01DC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13	Reserved	RSVD 0x0	Reserved
12	PU_PLL150M	RW 0x1	Pull up PLL150M
11:0	Reserved	RSVD 0x780	Reserved

Copyright © 2015 Marvell

Table 1601: Cal_reg19 Register (n=0–6)

Offset: Port0: 0x000A01E0, Port1: 0x000A09E0, Port2: 0x000A11E0, Port3: 0x000A19E0, Port4: 0x000A21E0, Port5: 0x000A29E0,

Offset Formula: 0x000A01E0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:14	Reserved	RSVD 0x0	Reserved
13:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Table 1602: Impedance Calibration Result Register (n=0–6)

Offset: Port0: 0x000A01E4, Port1: 0x000A09E4, Port2: 0x000A11E4, Port3: 0x000A19E4, Port4: 0x000A21E4, Port5: 0x000A29E4,

Offset Formula: 0x000A01E4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved
5:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1603: Impedance Calibration Result Register (n=0–6)

Offset: Port0: 0x000A01E8, Port1: 0x000A09E8, Port2: 0x000A11E8, Port3: 0x000A19E8, Port4: 0x000A21E8, Port5: 0x000A29E8,

Offset Formula: 0x000A01E8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved
5:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1604: F1 Calibration Result Register (n=0–6)

Offset: Port0: 0x000A01EC, Port1: 0x000A09EC, Port2: 0x000A11EC, Port3: 0x000A19EC, Port4: 0x000A21EC, Port5: 0x000A29EC,

Offset Formula: 0x000A01EC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved
5:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1605: RX DCC Calibration Binary Register (n=0–6)

Offset: Port0: 0x000A01F0, Port1: 0x000A09F0, Port2: 0x000A11F0, Port3: 0x000A19F0, Port4: 0x000A21F0, Port5: 0x000A29F0,

Offset Formula: 0x000A01F0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x0	Reserved

Table 1606: TX DCC Calibration Binary Register (n=0–6)

Offset: Port0: 0x000A01F4, Port1: 0x000A09F4, Port2: 0x000A11F4, Port3: 0x000A19F4, Port4: 0x000A21F4, Port5: 0x000A29F4,

Offset Formula: 0x000A01F4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:6	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
5:0	Reserved	RSVD 0x0	Reserved

Table 1607: Squelch Control 2 Register (n=0–6)

Offset: Port0: 0x000A01F8, Port1: 0x000A09F8, Port2: 0x000A11F8, Port3: 0x000A19F8, Port4: 0x000A21F8, Port5: 0x000A29F8,

Offset Formula: 0x000A01F8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x7	Reserved

Table 1608: RX Control 7 Register (n=0–6)

Offset: Port0: 0x000A0200, Port1: 0x000A0A00, Port2: 0x000A1200, Port3: 0x000A1A00, Port4: 0x000A2200, Port5: 0x000A2A00,

Offset Formula: 0x000A0200+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:2	Reserved	RSVD 0x1355	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1608: RX Control 7 Register (n=0–6) (Continued)

Offset: Port0: 0x000A0200, Port1: 0x000A0A00, Port2: 0x000A1200, Port3: 0x000A1A00, Port4: 0x000A2200, Port5: 0x000A2A00,

Offset Formula: 0x000A0200+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1609: AC JTAG Control 0 Register (n=0–6)

Offset: Port0: 0x000A0204, Port1: 0x000A0A04, Port2: 0x000A1204, Port3: 0x000A1A04, Port4: 0x000A2204, Port5: 0x000A2A04,

Offset Formula: 0x000A0204+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:0	Reserved	RSVD 0xA1	Reserved

Table 1610: DFE F1p5 and F2p5 Control and Other Register (n=0–6)

Offset: Port0: 0x000A0208, Port1: 0x000A0A08, Port2: 0x000A1208, Port3: 0x000A1A08, Port4: 0x000A2208, Port5: 0x000A2A08,

Offset Formula: 0x000A0208+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	FFE_OS_INV	RW 0x0	invert ffe_os bit 3
14:11	Reserved	RSVD 0x0	Reserved
10:9	DFE_FXP5_FORCE [1:0]	RW 0x0	Force dfe_f1p5 and dfe_f2p5 [0]: force dfe_f1p5 [1]: force dfe_f2p5
8:5	DFE_F2P5_EXT[3:0]	RW 0x0	Forced dfe_f2p5
4:0	DFE_F1P5_EXT[4:0]	RW 0x0	Forced dfe_f1p5

Table 1611: FFE Control Register (n=0–6)

Offset: Port0: 0x000A020C, Port1: 0x000A0A0C, Port2: 0x000A120C, Port3: 0x000A1A0C, Port4: 0x000A220C, Port5: 0x000A2A0C,

Offset Formula: 0x000A020C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	Reserved	RSVD 0x0	Reserved

Table 1611: FFE Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A020C, Port1: 0x000A0A0C, Port2: 0x000A120C, Port3: 0x000A1A0C, Port4: 0x000A220C, Port5: 0x000A2A0C,

Offset Formula: 0x000A020C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
9	FFE_OS_FORCE	RW 0x0	force FFE_OS to use reg_ffe_os
8	EDGE_EQ_EN	RW 0x1	Edge equalization is enabled If edge_eq_en is high, dfe_f1p5, dfe_f2p5 and ffe_os are adapted.
7:6	Reserved	RSVD 0x0	Reserved
5:1	FFE_OS_EXT[4:0]	RW 0x0	Signed 5 bits outputs to control FFE DC offset control. It is only used when reg_ffe_os_force is high
0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1612: DFE 14 Register (n=0–6)

Offset: Port0: 0x000A0210, Port1: 0x000A0A10, Port2: 0x000A1210, Port3: 0x000A1A10, Port4: 0x000A2210, Port5: 0x000A2A10,

Offset Formula: 0x000A0210+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0xE20A	Reserved

Table 1613: Frame Detection Control 0 Register (n=0–6)

Offset: Port0: 0x000A0214, Port1: 0x000A0A14, Port2: 0x000A1214, Port3: 0x000A1A14, Port4: 0x000A2214, Port5: 0x000A2A14,

Offset Formula: 0x000A0214+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:3	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
2	Reserved	RSVD 0x0	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1614: PLL Calibration Control 11 Register (n=0–6)

Offset: Port0: 0x000A0218, Port1: 0x000A0A18, Port2: 0x000A1218, Port3: 0x000A1A18, Port4: 0x000A2218, Port5: 0x000A2A18,

Offset Formula: 0x000A0218+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:11	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1614: PLL Calibration Control 11 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0218, Port1: 0x000A0A18, Port2: 0x000A1218, Port3: 0x000A1A18, Port4: 0x000A2218, Port5: 0x000A2A18,
Offset Formula: 0x000A0218+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
10:4	Reserved	RSVD 0x40	Reserved
3:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1615: Frame Detection Control 2 Register (n=0–6)
Offset: Port0: 0x000A021C, Port1: 0x000A0A1C, Port2: 0x000A121C, Port3: 0x000A1A1C, Port4: 0x000A221C, Port5: 0x000A2A1C,
Offset Formula: 0x000A021C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1616: Frame Detection Control 3 Register (n=0–6)
Offset: Port0: 0x000A0220, Port1: 0x000A0A20, Port2: 0x000A1220, Port3: 0x000A1A20, Port4: 0x000A2220, Port5: 0x000A2A20,
Offset Formula: 0x000A0220+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	Reserved	RSVD 0x1	Reserved
9:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3:0	Reserved	RSVD 0x0	Reserved

Table 1617: TX Training Control 0 Register (n=0–6)
Offset: Port0: 0x000A022C, Port1: 0x000A0A2C, Port2: 0x000A122C, Port3: 0x000A1A2C, Port4: 0x000A222C, Port5: 0x000A2A2C,
Offset Formula: 0x000A022C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x4	Reserved

Copyright © 2015 Marvell

Table 1617: TX Training Control 0 Register (n=0–6) (Continued)

Offset: Port0: 0x000A022C, Port1: 0x000A0A2C, Port2: 0x000A122C, Port3: 0x000A1A2C, Port4: 0x000A222C, Port5: 0x000A2A2C,

Offset Formula: 0x000A022C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11	RX_TEST_FFE_MODE	RW 0x0	FFE test mode If Test_FFE_mode==1, use RX training stage 3 and train a set of given FFEs as in 6G SerDes design. Otherwise, do the regular RX training with all possible stages
10	Reserved	RSVD 0x0	Reserved
9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8	TX_NO_INIT	RW 0x0	tx training without reset TX FFE settings TX_NO_INIT = 1: we will not send out TX preset commands to remote PHY in the beginning of TX training. TX_NO_INIT = 0: we will send out TX preset commands to remote PHY. Which preset command is determined by TX_PRESET_INDEX.
7	RX_NO_INIT	RW 0x0	rx training without reset rx FFE settings RX_NO_INIT = 1: we will not reset RX FFE setting in the beginning of TX/RX training. RX_NO_INIT = 0: we will reset RX FFE setting to r_default and c_default in the beginning of TX/RX training.
6:2	Reserved	RSVD 0x1C	Reserved
1	TX_TRAIN_ENABLE	RW 0x0	TX training enable. In isolate mode, this bit is used instead of PIN_TX_TRAIN_ENABLE. In non isolate mode, this bit is Ored with PIN_TX_TRAIN_ENABLE.
0	Reserved	RSVD 0x0	Reserved

Table 1618: TRX Training Control 1 Register (n=0–6)

Offset: Port0: 0x000A0230, Port1: 0x000A0A30, Port2: 0x000A1230, Port3: 0x000A1A30, Port4: 0x000A2230, Port5: 0x000A2A30,

Offset Formula: 0x000A0230+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	PIN_TRAIN_COMPLETE_TYPE	RW 0x0	PIN_TX_TRAIN_COMPELTE and PIN_RX_TRAIN_COMPLETE type 0 = deassert PIN_TX_TRAIN_COMPLETE, PIN_RX_TRAIN_COMPLETE, PIN_TX_TRAIN_FAILED and PIN_RX_TRAIN_FAILED when register TX_TRAIN_ENABLE is low 1 = PIN_TX_TRAIN_COMPLETE, PIN_RX_TRAIN_COMPLETE, PIN_TX_TRAIN_FAILED and PIN_RX_TRAIN_COMPLETE only have one clock cycle

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1618: TRX Training Control 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0230, Port1: 0x000A0A30, Port2: 0x000A1230, Port3: 0x000A1A30, Port4: 0x000A2230, Port5: 0x000A2A30,
Offset Formula: 0x000A0230+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
14:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:0	Reserved	RSVD 0x280	Reserved

Table 1619: RX Training Control 0 Register (n=0–6)
Offset: Port0: 0x000A0234, Port1: 0x000A0A34, Port2: 0x000A1234, Port3: 0x000A1A34, Port4: 0x000A2234, Port5: 0x000A2A34,
Offset Formula: 0x000A0234+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1000	Reserved

Table 1620: RX Training Control 1 Register (n=0–6)
Offset: Port0: 0x000A0238, Port1: 0x000A0A38, Port2: 0x000A1238, Port3: 0x000A1A38, Port4: 0x000A2238, Port5: 0x000A2A38,
Offset Formula: 0x000A0238+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x1000	Reserved

Table 1621: RX Training Control 2 Register (n=0–6)
Offset: Port0: 0x000A023C, Port1: 0x000A0A3C, Port2: 0x000A123C, Port3: 0x000A1A3C, Port4: 0x000A223C, Port5: 0x000A2A3C,
Offset Formula: 0x000A023C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1000	Reserved

Table 1622: RX Training Control 3 Register (n=0–6)
Offset: Port0: 0x000A0240, Port1: 0x000A0A40, Port2: 0x000A1240, Port3: 0x000A1A40, Port4: 0x000A2240, Port5: 0x000A2A40,
Offset Formula: 0x000A0240+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:4	Reserved	RSVD 0xC	Reserved

Copyright © 2015 Marvell

Table 1622: RX Training Control 3 Register (n=0–6) (Continued)

Offset: Port0: 0x000A0240, Port1: 0x000A0A40, Port2: 0x000A1240, Port3: 0x000A1A40, Port4: 0x000A2240, Port5: 0x000A2A40,

Offset Formula: 0x000A0240+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3	CDRPHASE_OPT_EN	RW 0x1	CDR phase optimization enable
2	CDRPHASE_OPT_ONLY_IN_LAST	RW 0x1	CDR phase optimization only in last DFE adaptation in RX training
1:0	Reserved	RSVD 0x2	Reserved

Table 1623: RX Training Control 4 Register (n=0–6)

Offset: Port0: 0x000A0244, Port1: 0x000A0A44, Port2: 0x000A1244, Port3: 0x000A1A44, Port4: 0x000A2244, Port5: 0x000A2A44,

Offset Formula: 0x000A0244+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0xF	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x1F	Reserved

Table 1624: RX Training Control 5 Register (n=0–6)

Offset: Port0: 0x000A0248, Port1: 0x000A0A48, Port2: 0x000A1248, Port3: 0x000A1A48, Port4: 0x000A2248, Port5: 0x000A2A48,

Offset Formula: 0x000A0248+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x2F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x3F	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1625: RX Training Control 6 Register (n=0–6)

Offset: Port0: 0x000A024C, Port1: 0x000A0A4C, Port2: 0x000A124C, Port3: 0x000A1A4C, Port4: 0x000A224C, Port5: 0x000A2A4C,

Offset Formula: 0x000A024C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x4F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x5F	Reserved

Table 1626: RX Training Control 7 Register (n=0–6)

Offset: Port0: 0x000A0250, Port1: 0x000A0A50, Port2: 0x000A1250, Port3: 0x000A1A50, Port4: 0x000A2250, Port5: 0x000A2A50,

Offset Formula: 0x000A0250+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x6F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x7F	Reserved

Table 1627: RX Training Control 8 Register (n=0–6)

Offset: Port0: 0x000A0254, Port1: 0x000A0A54, Port2: 0x000A1254, Port3: 0x000A1A54, Port4: 0x000A2254, Port5: 0x000A2A54,

Offset Formula: 0x000A0254+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x7	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x27	Reserved

Copyright © 2015 Marvell

Table 1628: RX Training Control 9 Register (n=0–6)

Offset: Port0: 0x000A0258, Port1: 0x000A0A58, Port2: 0x000A1258, Port3: 0x000A1A58, Port4: 0x000A2258, Port5: 0x000A2A58,

Offset Formula: 0x000A0258+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x47	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x67	Reserved

Table 1629: RX Training Control 10 Register (n=0–6)

Offset: Port0: 0x000A025C, Port1: 0x000A0A5C, Port2: 0x000A125C, Port3: 0x000A1A5C, Port4: 0x000A225C, Port5: 0x000A2A5C,

Offset Formula: 0x000A025C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x1	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x21	Reserved

Table 1630: RX Training Control 11 Register (n=0–6)

Offset: Port0: 0x000A0260, Port1: 0x000A0A60, Port2: 0x000A1260, Port3: 0x000A1A60, Port4: 0x000A2260, Port5: 0x000A2A60,

Offset Formula: 0x000A0260+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x41	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1630: RX Training Control 11 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0260, Port1: 0x000A0A60, Port2: 0x000A1260, Port3: 0x000A1A60, Port4: 0x000A2260, Port5: 0x000A2A60,
Offset Formula: 0x000A0260+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
6:0	Reserved	RSVD 0x61	Reserved

Table 1631: RX Training Control 12 Register (n=0–6)
Offset: Port0: 0x000A0264, Port1: 0x000A0A64, Port2: 0x000A1264, Port3: 0x000A1A64, Port4: 0x000A2264, Port5: 0x000A2A64,
Offset Formula: 0x000A0264+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	Reserved	RSVD 0x1	Reserved
13:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1632: TX Training Control 0 Register (n=0–6)
Offset: Port0: 0x000A0268, Port1: 0x000A0A68, Port2: 0x000A1268, Port3: 0x000A1A68, Port4: 0x000A2268, Port5: 0x000A2A68,
Offset Formula: 0x000A0268+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	TX_TRAIN_P2P_HOLD	RW 0x1	TX train peak to peak hold enable 0 = TX coefficient control separately 1 = TX g0 control same as g1 or gn1 control
14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:11	Reserved	RSVD 0x1	Reserved
10:2	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
1:0	Reserved	RSVD 0x2	Reserved

Table 1633: TX Training Control 1 Register (n=0–6)
Offset: Port0: 0x000A026C, Port1: 0x000A0A6C, Port2: 0x000A126C, Port3: 0x000A1A6C, Port4: 0x000A226C, Port5: 0x000A2A6C,
Offset Formula: 0x000A026C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x6B90	Reserved

Copyright © 2015 Marvell

Table 1634: TRX Training Control 2 Register (n=0–6)

Offset: Port0: 0x000A0270, Port1: 0x000A0A70, Port2: 0x000A1270, Port3: 0x000A1A70, Port4: 0x000A2270, Port5: 0x000A2A70,

Offset Formula: 0x000A0270+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	EYE_CHECK_DISABLE	RW 0x0	Force eye check always pass in TX/RX training This bit will not disable eye check state in state machine 0 = eye_check_pass will be used in eye open result 1 = eye_check_pass will not be used in eye open result
14:5	Reserved	RSVD 0x40	Reserved
4:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1635: TRX Training Control 4 Register (n=0–6)

Offset: Port0: 0x000A0278, Port1: 0x000A0A78, Port2: 0x000A1278, Port3: 0x000A1A78, Port4: 0x000A2278, Port5: 0x000A2A78,

Offset Formula: 0x000A0278+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:11	Reserved	RSVD 0x4	Reserved
10	TRX_TRAIN_TIMEOUT_EN	RW 0x1	TX/RX training timeout enable If this bit is set to high, TX/RX training will timeout when training time is more than TRX_TRAIN_TIMER or RX_TRAIN_TIMER. If timeout, TX/RX_TRAIN_FAILED is high.
9:0	TRX_TRAIN_TIMER[9:0]	RW 0x1F3	TX trainning maximum time unit is 1ms

Table 1636: DFE Adaptation Control 0 Register (n=0–6)

Offset: Port0: 0x000A027C, Port1: 0x000A0A7C, Port2: 0x000A127C, Port3: 0x000A1A7C, Port4: 0x000A227C, Port5: 0x000A2A7C,

Offset Formula: 0x000A027C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x9651	Reserved

Copyright © 2015 Marvell

Table 1637: DFE Adaptation Control 1 Register (n=0–6)

Offset: Port0: 0x000A0280, Port1: 0x000A0A80, Port2: 0x000A1280, Port3: 0x000A1A80, Port4: 0x000A2280, Port5: 0x000A2A80,

Offset Formula: 0x000A0280+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x80BC	Reserved

Table 1638: DFE Adaptation Control 2 Register (n=0–6)

Offset: Port0: 0x000A0284, Port1: 0x000A0A84, Port2: 0x000A1284, Port3: 0x000A1A84, Port4: 0x000A2284, Port5: 0x000A2A84,

Offset Formula: 0x000A0284+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x5800	Reserved

Table 1639: TX Amplitude Setting for PIPE Register (n=0–6)

Offset: Port0: 0x000A0288, Port1: 0x000A0A88, Port2: 0x000A1288, Port3: 0x000A1A88, Port4: 0x000A2288, Port5: 0x000A2A88,

Offset Formula: 0x000A0288+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x0	Reserved
11:7	tx_amp_pipe_v0[4:0]	RW 0x16	TX amplitude for TX margin is 0 default is 1000mv
6:0	Reserved	RSVD 0x20	Reserved

Table 1640: TX Margin Selection Register (n=0–6)

Offset: Port0: 0x000A0290, Port1: 0x000A0A90, Port2: 0x000A1290, Port3: 0x000A1A90, Port4: 0x000A2290, Port5: 0x000A2A90,

Offset Formula: 0x000A0290+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	tx_emph_pipe1[3:0]	RW 0xA	tx_emph value for -3.5db
15:8	Reserved	RSVD 0xAF	Reserved
11:8	tx_emph_pipe0[3:0]	RW 0xF	tx_emph value for -6db
7:5	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1640: TX Margin Selection Register (n=0–6) (Continued)
Offset: Port0: 0x000A0290, Port1: 0x000A0A90, Port2: 0x000A1290, Port3: 0x000A1A90, Port4: 0x000A2290, Port5: 0x000A2A90,
Offset Formula: 0x000A0290+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
4:0	Reserved	RSVD 0x17	Reserved

Table 1641: Tx_Amp_Adj and Tx_Emph_En Register (n=0–6)
Offset: Port0: 0x000A0294, Port1: 0x000A0A94, Port2: 0x000A1294, Port3: 0x000A1A94, Port4: 0x000A2294, Port5: 0x000A2A94,
Offset Formula: 0x000A0294+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12	Reserved	RSVD 0x1	Reserved
11:8	TX_EMPH0_MAX[3:0]	RW 0xE	TX emphasis 0 maximum index used for TX training
7:4	TX_EMPH1_MAX[3:0]	RW 0xE	TX emphasis 1 maximum index used for TX training
3:0	Reserved	RSVD 0x4	Reserved

Table 1642: TX Amplitude Index Limit Register (n=0–6)
Offset: Port0: 0x000A029C, Port1: 0x000A0A9C, Port2: 0x000A129C, Port3: 0x000A1A9C, Port4: 0x000A229C, Port5: 0x000A2A9C,
Offset Formula: 0x000A029C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:6	TX_AMP_MIN[5:0]	RW 0x8	TX amplitude mininum index used for TX training
5:0	TX_AMP_MAX[5:0]	RW 0x1C	TX amplitude maximum index used for TX training

Copyright © 2015 Marvell

Table 1643: TX Emphasis Index Minimum Value Register (n=0–6)

Offset: Port0: 0x000A02A0, Port1: 0x000A0AA0, Port2: 0x000A12A0, Port3: 0x000A1AA0, Port4: 0x000A22A0, Port5: 0x000A2AA0,

Offset Formula: 0x000A02A0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:8	TX_TRAIN_START_WAIT_TIME[1:0]	RW 0x0	TX training start wait time When TX training is enabled, wait a certain time to start training. 0 = 0.5ms 1 = V1ms: 1ms 2 = 1.5ms 3 = V2ms: 2ms
7:4	TX_EMPH1_MIN[3:0]	RW 0x0	TX emphasis 0 minimum index used for TX training
3:0	TX_EMPH0_MIN[3:0]	RW 0x0	TX emphasis 1 minimum index used for TX training

Table 1644: TRX Training Control 5 Register (n=0–6)

Offset: Port0: 0x000A02A4, Port1: 0x000A0AA4, Port2: 0x000A12A4, Port3: 0x000A1AA4, Port4: 0x000A22A4, Port5: 0x000A2AA4,

Offset Formula: 0x000A02A4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14	TX_TRAIN_START_WAIT_TIME_EN	RW 0x1	Start TX training when wait time is reached.
13:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11	TX_TRAIN_START_SQ_EN	RW 0x1	Start TX training when SQ_DETECTED is low
10	TX_TRAIN_START_NO_WAIT	RW 0x0	Start TX training immediately
9:0	RX_TRAIN_TIMER[9:0]	RW 0x13	RX train maximum timer unit is 1ms

Table 1645: Interrupt 1 Register (n=0–6)

Offset: Port0: 0x000A02AC, Port1: 0x000A0AAC, Port2: 0x000A12AC, Port3: 0x000A1AAC, Port4: 0x000A22AC, Port5: 0x000A2AAC,

Offset Formula: 0x000A02AC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1645: Interrupt 1 Register (n=0–6) (Continued)

Offset: Port0: 0x000A02AC, Port1: 0x000A0AAC, Port2: 0x000A12AC, Port3: 0x000A1AAC, Port4: 0x000A22AC, Port5: 0x000A2AAC,

Offset Formula: 0x000A02AC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
6	TRX_TRAIN_FAILED_INT	RO 0x0	TX/RX trainining failed interrupt
5	TRX_TRAIN_TIMEOUT_INT	RO 0x0	TX/RX training timeout interrupt. High means TX/RX traiing timeout.
4	TRX_TRAIN_DONE_INT	RO 0x0	TX/RX training done interrupt. High menas TRX training state machine is finished.
3	DFE_DONE_INT	RO 0x0	DFE done interrupt. High means one DFE adaptation in TX/RX training is done.
2	STATUS_DET_TIMEOUT_INT	RO 0x0	Status detection timeout interrupt. High means local PHY cannot get correct coefficient status from remote PHY.
1	RX_TRAIN_COMPLETE_INT	RO 0x0	RX train complete interrupt. High means RX traning is done.
0	TX_TRAIN_COMPLETE_INT	RO 0x0	TX train complete interrupt. High means TX training is done. If TX training is failed, this interrupt is also asserted.

Table 1646: Interrupt 3 Register (n=0–6)

Offset: Port0: 0x000A02B4, Port1: 0x000A0AB4, Port2: 0x000A12B4, Port3: 0x000A1AB4, Port4: 0x000A22B4, Port5: 0x000A2AB4,

Offset Formula: 0x000A02B4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6	TRX_TRAIN_FAILED_INT_EN	RW 0x0	TRX training failed interrupt enable
5	TRX_TRAIN_TIMEOUT_INT_EN	RW 0x0	TRX training timeout interrupt enable.
4	TRX_TRAIN_DONE_INT_EN	RW 0x0	TRX training done interrupt enable.
3	DFE_DONE_INT_EN	RW 0x0	DFE done interrupt enable
2	STATUS_DET_TIMEOUT_INT_EN	RW 0x0	Status detection timeout interrupt enable.
1	RX_TRAIN_COMPLETE_INT_EN	RW 0x0	RX train complete interrupt enable
0	TX_TRAIN_COMPLETE_INT_EN	RW 0x0	TX train complete interrupt enable

Copyright © 2015 Marvell

Table 1647: Interrupt 5 Register (n=0–6)

Offset: Port0: 0x000A02BC, Port1: 0x000A0ABC, Port2: 0x000A12BC, Port3: 0x000A1ABC, Port4: 0x000A22BC, Port5: 0x000A2ABC,

Offset Formula: 0x000A02BC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6	TRX_TRAIN_FAILED_INT_CLEAR	RW 0x0	TX/RX training failed interrupt clear
5	TRX_TRAIN_TIMEOUT_INT_CLEAR	RW 0x0	TRX training timeout interrupt clear.
4	TRX_TRAIN_DONE_INT_CLEAR	RW 0x0	TRX training done interrupt clear.
3	DFE_DONE_INT_CLEAR	RW 0x0	DFE done interrupt clear
2	STATUS_DET_TIMEOUT_INT_CLEAR	RW 0x0	Status detection timeout interrupt clear.
1	RX_TRAIN_COMPLETE_INT_CLEAR	RW 0x0	RX train complete interrupt clear
0	TX_TRAIN_COMPLETE_INT_CLEAR	RW 0x0	TX train complete interrupt clear

Table 1648: TX Amplitude Default Register (n=0–6)

Offset: Port0: 0x000A02DC, Port1: 0x000A0ADC, Port2: 0x000A12DC, Port3: 0x000A1ADC, Port4: 0x000A22DC, Port5: 0x000A2ADC,

Offset Formula: 0x000A02DC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0xE80	Reserved

Table 1649: Value of Signals to Analog Selection Register (n=0–6)

Offset: Port0: 0x000A02E0, Port1: 0x000A0AE0, Port2: 0x000A12E0, Port3: 0x000A1AE0, Port4: 0x000A22E0, Port5: 0x000A2AE0,

Offset Formula: 0x000A02E0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:8	ANA_VIRTUAL_CTRL[7:0]	RW 0x0	Controls the ANA_VIRTUAL_CTRL[7:0] ports of the analog design. These ports have no real circuit function, but are used in the analog behavioral model for simulation purposes.
7:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1650: EOM Phase Control Register (n=0–6)

Offset: Port0: 0x000A02EC, Port1: 0x000A0AEC, Port2: 0x000A12EC, Port3: 0x000A1AEC, Port4: 0x000A22EC, Port5: 0x000A2AEC,

Offset Formula: 0x000A02EC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8:0	Reserved	RSVD 0x60	Reserved

Table 1651: EOM Calibration Control 0 Register (n=0–6)

Offset: Port0: 0x000A0308, Port1: 0x000A0B08, Port2: 0x000A1308, Port3: 0x000A1B08, Port4: 0x000A2308, Port5: 0x000A2B08,

Offset Formula: 0x000A0308+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0x300	Reserved
3:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1652: EOM Calibration Control 1 Register (n=0–6)

Offset: Port0: 0x000A030C, Port1: 0x000A0B0C, Port2: 0x000A130C, Port3: 0x000A1B0C, Port4: 0x000A230C, Port5: 0x000A2B0C,

Offset Formula: 0x000A030C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1653: TX Train Result Register (n=0–6)

Offset: Port0: 0x000A0310, Port1: 0x000A0B10, Port2: 0x000A1310, Port3: 0x000A1B10, Port4: 0x000A2310, Port5: 0x000A2B10,

Offset Formula: 0x000A0310+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1654: TX Train Result Register (n=0–6)

Offset: Port0: 0x000A0314, Port1: 0x000A0B14, Port2: 0x000A1314, Port3: 0x000A1B14, Port4: 0x000A2314, Port5: 0x000A2B14,

Offset Formula: 0x000A0314+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7	Reserved	RSVD 0x0	Reserved
6	TX_TRAIN_COMPLETE	RO 0x0	TX training complete status
5	TX_TRAIN_FAILED	RO 0x0	TX training failed status It is in TXCLK domain
4	RX_TRAIN_COMPLETE	RO 0x0	RX training complete status
3	RX_TRAIN_FAILED	RO 0x0	RX training falied status It is in TXCLK domain
2:1	TX_TRAIN_ERROR[1:0]	RO 0x0	TX train error information 0 = pattern lock lost timer expires(only in auto mode) 1 = there are no additional, untried, commonly supported settings for local PHY 2 = MTTT timer expires for local PHY 3 = does not get remote PHY's complete status within MTTT timer
0	TRX_TRAIN_TIMEOUT	RO 0x0	TX/RX training timeout flag

Table 1655: Adapted Edge Result Register (n=0–6)

Offset: Port0: 0x000A0318, Port1: 0x000A0B18, Port2: 0x000A1318, Port3: 0x000A1B18, Port4: 0x000A2318, Port5: 0x000A2B18,

Offset Formula: 0x000A0318+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:10	DFE_F1P5[4:0]	RO 0x0	adapted f1p5
9:6	DFE_F2P5[3:0]	RO 0x0	adapted f2p5
5:1	FFE_OS[4:0]	RO 0x0	adapted ffe_os
0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1656: TX Train Preset Register (n=0–6)

Offset: Port0: 0x000A031C, Port1: 0x000A0B1C, Port2: 0x000A131C, Port3: 0x000A1B1C, Port4: 0x000A231C, Port5: 0x000A2B1C,

Offset Formula: 0x000A031C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:11	Reserved	RSVD 0x0	Reserved
10:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7	TX_COE_FM_PIN_PCIE3_EN	RW 0x1	TX coefficient from pin enable for PCIE3 mode It is used in PCIE3 mode. 0 = TX coefficient is from tx_margin tx_deemph 1 = TX coefficient is from PIN_RESERVED_INPUT_P[17:0]
6:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3:2	Reserved	RSVD 0x2	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1657: Phase Offset Range Register (n=0–6)

Offset: Port0: 0x000A0320, Port1: 0x000A0B20, Port2: 0x000A1320, Port3: 0x000A1B20, Port4: 0x000A2320, Port5: 0x000A2B20,

Offset Formula: 0x000A0320+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:1	Reserved	RSVD 0xB68	Reserved
0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1658: Saved DFE Values 0 Register (n=0–6)

Offset: Port0: 0x000A0324, Port1: 0x000A0B24, Port2: 0x000A1324, Port3: 0x000A1B24, Port4: 0x000A2324, Port5: 0x000A2B24,

Offset Formula: 0x000A0324+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1659: Saved DFE Values 1 Register (n=0–6)

Offset: Port0: 0x000A0328, Port1: 0x000A0B28, Port2: 0x000A1328, Port3: 0x000A1B28, Port4: 0x000A2328, Port5: 0x000A2B28,

Offset Formula: 0x000A0328+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1660: Saved DFE Values 2 Register (n=0–6)

Offset: Port0: 0x000A032C, Port1: 0x000A0B2C, Port2: 0x000A132C, Port3: 0x000A1B2C, Port4: 0x000A232C, Port5: 0x000A2B2C,

Offset Formula: 0x000A032C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x8400	Reserved

Table 1661: RX Train Default Values Register (n=0–6)

Offset: Port0: 0x000A0330, Port1: 0x000A0B30, Port2: 0x000A1330, Port3: 0x000A1B30, Port4: 0x000A2330, Port5: 0x000A2B30,

Offset Formula: 0x000A0330+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9	Reserved	RSVD 0x0	Reserved
8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:0	Reserved	RSVD 0xF	Reserved

Table 1662: Status Detection Timeout Control Register (n=0–6)

Offset: Port0: 0x000A0334, Port1: 0x000A0B34, Port2: 0x000A1334, Port3: 0x000A1B34, Port4: 0x000A2334, Port5: 0x000A2B34,

Offset Formula: 0x000A0334+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13	Reserved	RSVD 0x0	Reserved
12:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1662: Status Detection Timeout Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A0334, Port1: 0x000A0B34, Port2: 0x000A1334, Port3: 0x000A1B34, Port4: 0x000A2334, Port5: 0x000A2B34,

Offset Formula: 0x000A0334+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
9:0	Reserved	RSVD 0x103	Reserved

Table 1663: TX Amplitude Default Register (n=0–6)

Offset: Port0: 0x000A0338, Port1: 0x000A0B38, Port2: 0x000A1338, Port3: 0x000A1B38, Port4: 0x000A2338, Port5: 0x000A2B38,

Offset Formula: 0x000A0338+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1150	Reserved

Table 1664: TX Amplitude Default Register (n=0–6)

Offset: Port0: 0x000A033C, Port1: 0x000A0B3C, Port2: 0x000A133C, Port3: 0x000A1B3C, Port4: 0x000A233C, Port5: 0x000A2B3C,

Offset Formula: 0x000A033C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1060	Reserved

Table 1665: DFE Adaptation Control 3 Register (n=0–6)

Offset: Port0: 0x000A0340, Port1: 0x000A0B40, Port2: 0x000A1340, Port3: 0x000A1B40, Port4: 0x000A2340, Port5: 0x000A2B40,

Offset Formula: 0x000A0340+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x4E93	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1666: RX Training Control 14 Register (n=0–6)

Offset: Port0: 0x000A0348, Port1: 0x000A0B48, Port2: 0x000A1348, Port3: 0x000A1B48, Port4: 0x000A2348, Port5: 0x000A2B48,

Offset Formula: 0x000A0348+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1000	Reserved

Table 1667: RX Training Control 15 Register (n=0–6)

Offset: Port0: 0x000A034C, Port1: 0x000A0B4C, Port2: 0x000A134C, Port3: 0x000A1B4C, Port4: 0x000A234C, Port5: 0x000A2B4C,

Offset Formula: 0x000A034C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1000	Reserved

Table 1668: RX Training Control 16 Register (n=0–6)

Offset: Port0: 0x000A0350, Port1: 0x000A0B50, Port2: 0x000A1350, Port3: 0x000A1B50, Port4: 0x000A2350, Port5: 0x000A2B50,

Offset Formula: 0x000A0350+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1669: TX Training Control 6 Register (n=0–6)

Offset: Port0: 0x000A035C, Port1: 0x000A0B5C, Port2: 0x000A135C, Port3: 0x000A1B5C, Port4: 0x000A235C, Port5: 0x000A2B5C,

Offset Formula: 0x000A035C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x5E01	Reserved

Table 1670: TX Training Control 7 Register (n=0–6)

Offset: Port0: 0x000A0360, Port1: 0x000A0B60, Port2: 0x000A1360, Port3: 0x000A1B60, Port4: 0x000A2360, Port5: 0x000A2B60,

Offset Formula: 0x000A0360+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0xFFC	Reserved
3:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1671: TX Training Control 7 Register (n=0–6)

Offset: Port0: 0x000A0364, Port1: 0x000A0B64, Port2: 0x000A1364, Port3: 0x000A1B64, Port4: 0x000A2364, Port5: 0x000A2B64,

Offset Formula: 0x000A0364+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0xFFC	Reserved
3:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1672: TRX Training Control 6 Register (n=0–6)

Offset: Port0: 0x000A036C, Port1: 0x000A0B6C, Port2: 0x000A136C, Port3: 0x000A1B6C, Port4: 0x000A236C, Port5: 0x000A2B6C,

Offset Formula: 0x000A036C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x0	Reserved
11	DFE_RES_TRAIN_END_EN	RW 0x1	DFE resolution training enable during TRX Train End State
10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9	DFE_RES_TRAIN_INIT_EN	RW 0x1	DFE resolution training enable during TRX Train Init State
8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:0	Reserved	RSVD 0xF8	Reserved

Table 1673: Calibration Control 16 Register (n=0–6)

Offset: Port0: 0x000A0370, Port1: 0x000A0B70, Port2: 0x000A1370, Port3: 0x000A1B70, Port4: 0x000A2370, Port5: 0x000A2B70,

Offset Formula: 0x000A0370+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:2	Reserved	RSVD 0x0	Reserved
1	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1674: VDD Calibration Control 4 Register (n=0–6)

Offset: Port0: 0x000A0374, Port1: 0x000A0B74, Port2: 0x000A1374, Port3: 0x000A1B74, Port4: 0x000A2374, Port5: 0x000A2B74,

Offset Formula: 0x000A0374+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8:0	Reserved	RSVD 0x41	Reserved

Table 1675: F1 Buffer Offset Calibration Control 0 Register (n=0–6)

Offset: Port0: 0x000A0378, Port1: 0x000A0B78, Port2: 0x000A1378, Port3: 0x000A1B78, Port4: 0x000A2378, Port5: 0x000A2B78,

Offset Formula: 0x000A0378+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x0	Reserved

Table 1676: TX Default 4 Register (n=0–6)

Offset: Port0: 0x000A037C, Port1: 0x000A0B7C, Port2: 0x000A137C, Port3: 0x000A1B7C, Port4: 0x000A237C, Port5: 0x000A2B7C,

Offset Formula: 0x000A037C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1240	Reserved

Table 1677: PLL Calibration Control 2 Register (n=0–6)

Offset: Port0: 0x000A0380, Port1: 0x000A0B80, Port2: 0x000A1380, Port3: 0x000A1B80, Port4: 0x000A2380, Port5: 0x000A2B80,

Offset Formula: 0x000A0380+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x82A0	Reserved

Table 1678: PLL Calibration Control 3 Register (n=0–6)

Offset: Port0: 0x000A0384, Port1: 0x000A0B84, Port2: 0x000A1384, Port3: 0x000A1B84, Port4: 0x000A2384, Port5: 0x000A2B84,

Offset Formula: 0x000A0384+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x572E	Reserved

Table 1679: PLL Calibration Control 4 Register (n=0–6)

Offset: Port0: 0x000A0388, Port1: 0x000A0B88, Port2: 0x000A1388, Port3: 0x000A1B88, Port4: 0x000A2388, Port5: 0x000A2B88,

Offset Formula: 0x000A0388+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:4	Reserved	RSVD 0x0	Reserved
3:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1680: PLL Calibration Control 5 Register (n=0–6)

Offset: Port0: 0x000A038C, Port1: 0x000A0B8C, Port2: 0x000A138C, Port3: 0x000A1B8C, Port4: 0x000A238C, Port5: 0x000A2B8C,

Offset Formula: 0x000A038C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1681: PLL Calibration Control 6 Register (n=0–6)

Offset: Port0: 0x000A0390, Port1: 0x000A0B90, Port2: 0x000A1390, Port3: 0x000A1B90, Port4: 0x000A2390, Port5: 0x000A2B90,

Offset Formula: 0x000A0390+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x7D	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1682: PLL Calibration Control 8 Register (n=0–6)

Offset: Port0: 0x000A0398, Port1: 0x000A0B98, Port2: 0x000A1398, Port3: 0x000A1B98, Port4: 0x000A2398, Port5: 0x000A2B98,

Offset Formula: 0x000A0398+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:7	Reserved	RSVD 0x20	Reserved
6	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
5	Reserved	RSVD 0x0	Reserved
4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3:0	Reserved	RSVD 0x3	Reserved

Table 1683: FFE Control 2 Register (n=0–6)

Offset: Port0: 0x000A039C, Port1: 0x000A0B9C, Port2: 0x000A139C, Port3: 0x000A1B9C, Port4: 0x000A239C, Port5: 0x000A2B9C,

Offset Formula: 0x000A039C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:6	Reserved	RSVD 0x15	Reserved
5:3	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
2:0	Reserved	RSVD 0x1	Reserved

Table 1684: Miscellaneous Control 3 Register (n=0–6)

Offset: Port0: 0x000A03A0, Port1: 0x000A0BA0, Port2: 0x000A13A0, Port3: 0x000A1BA0, Port4: 0x000A23A0, Port5: 0x000A2BA0,

Offset Formula: 0x000A03A0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0xD55	Reserved

Copyright © 2015 Marvell

Table 1685: Miscellaneous Control 4 Register (n=0–6)

Offset: Port0: 0x000A03A4, Port1: 0x000A0BA4, Port2: 0x000A13A4, Port3: 0x000A1BA4, Port4: 0x000A23A4, Port5: 0x000A2BA4,

Offset Formula: 0x000A03A4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Table 1686: TRX Training Control 7 Register (n=0–6)

Offset: Port0: 0x000A03A8, Port1: 0x000A0BA8, Port2: 0x000A13A8, Port3: 0x000A1BA8, Port4: 0x000A23A8, Port5: 0x000A2BA8,

Offset Formula: 0x000A03A8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x0	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1687: TRX Training Control 8 Register (n=0–6)

Offset: Port0: 0x000A03AC, Port1: 0x000A0BAC, Port2: 0x000A13AC, Port3: 0x000A1BAC, Port4: 0x000A23AC, Port5: 0x000A2BAC,

Offset Formula: 0x000A03AC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:0	Reserved	RSVD 0x2F00	Reserved

Table 1688: DFE Step Size 2 Register (n=0–6)

Offset: Port0: 0x000A03B0, Port1: 0x000A0BB0, Port2: 0x000A13B0, Port3: 0x000A1BB0, Port4: 0x000A23B0, Port5: 0x000A2BB0,

Offset Formula: 0x000A03B0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1688: DFE Step Size 2 Register (n=0–6) (Continued)
Offset: Port0: 0x000A03B0, Port1: 0x000A0BB0, Port2: 0x000A13B0, Port3: 0x000A1BB0, Port4: 0x000A23B0, Port5: 0x000A2BB0,
Offset Formula: 0x000A03B0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	DFE_STEP_COARSE_FXP5[3:0]	RW 0x2	dfe coarse step used for F1P5 and F2P5 measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
11:8	DFE_STEP_FINE_FXP5[3:0]	RW 0x6	dfe fine step used for F1P5 and F2P5 measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)
7:4	DFE_STEP_COARSE_OS[3:0]	RW 0x2	dfe coarse step used for FFE_OS measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Copyright © 2015 Marvell

Table 1688: DFE Step Size 2 Register (n=0–6) (Continued)

Offset: Port0: 0x000A03B0, Port1: 0x000A0BB0, Port2: 0x000A13B0, Port3: 0x000A1BB0, Port4: 0x000A23B0, Port5: 0x000A2BB0,

Offset Formula: 0x000A03B0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	DFE_STEP_FINE_OS[3:0]	RW 0x6	dfe fine step used for FFE_OS measurement 0 = 1 (coarsest step size) 1 = 1/2 2 = 1/4 3 = 1/8 4 = 1/16 5 = 1/32 6 = 1/64 7 = 1/128 8 = 1/256 9 = 1/512 10 = 1/1024 11 = 1/2048 (finest step size)

Table 1689: TRX Training Control 10 Register (n=0–6)

Offset: Port0: 0x000A03B4, Port1: 0x000A0BB4, Port2: 0x000A13B4, Port3: 0x000A1BB4, Port4: 0x000A23B4, Port5: 0x000A2BB4,

Offset Formula: 0x000A03B4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	Reserved	RSVD 0x0	Reserved
9:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:2	Reserved	RSVD 0x0	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1690: PLL Calibration Control Register (n=0–6)

Offset: Port0: 0x000A03B8, Port1: 0x000A0BB8, Port2: 0x000A13B8, Port3: 0x000A1BB8, Port4: 0x000A23B8, Port5: 0x000A2BB8,

Offset Formula: 0x000A03B8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	Reserved	RSVD 0x0	Reserved
8:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1691: Align 90 Calibration Control 0 Register (n=0–6)

Offset: Port0: 0x000A03BC, Port1: 0x000A0BBC, Port2: 0x000A13BC, Port3: 0x000A1BBC, Port4: 0x000A23BC, Port5: 0x000A2BBC,

Offset Formula: 0x000A03BC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:7	Reserved	RSVD 0x84	Reserved
6:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1692: Digital Test Bus Configuration Register (n=0–6)

Offset: Port0: 0x000A03C0, Port1: 0x000A0BC0, Port2: 0x000A13C0, Port3: 0x000A1BC0, Port4: 0x000A23C0, Port5: 0x000A2BC0,

Offset Formula: 0x000A03C0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	Reserved	RSVD 0x0	Reserved
14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:8	Reserved	RSVD 0x0	Reserved
7:6	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
5:0	Reserved	RSVD 0x0	Reserved

Table 1693: Digital Test Bus Register (n=0–6)

Offset: Port0: 0x000A03C4, Port1: 0x000A0BC4, Port2: 0x000A13C4, Port3: 0x000A1BC4, Port4: 0x000A23C4, Port5: 0x000A2BC4,

Offset Formula: 0x000A03C4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1694: Analog Control 1 Register (n=0–6)

Offset: Port0: 0x000A03C8, Port1: 0x000A0BC8, Port2: 0x000A13C8, Port3: 0x000A1BC8, Port4: 0x000A23C8, Port5: 0x000A2BC8,

Offset Formula: 0x000A03C8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:3	Reserved	RSVD 0x204	Reserved

Copyright © 2015 Marvell

Table 1694: Analog Control 1 Register (n=0–6) (Continued)
Offset: Port0: 0x000A03C8, Port1: 0x000A0BC8, Port2: 0x000A13C8, Port3: 0x000A1BC8, Port4: 0x000A23C8, Port5: 0x000A2BC8,
Offset Formula: 0x000A03C8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
2	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
1:0	Reserved	RSVD 0x0	Reserved

Table 1695: Analog Control 2 Register (n=0–6)
Offset: Port0: 0x000A03CC, Port1: 0x000A0BCC, Port2: 0x000A13CC, Port3: 0x000A1BCC, Port4: 0x000A23CC, Port5: 0x000A2BCC,
Offset Formula: 0x000A03CC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1696: Analog Control 3 Register (n=0–6)
Offset: Port0: 0x000A03D0, Port1: 0x000A0BD0, Port2: 0x000A13D0, Port3: 0x000A1BD0, Port4: 0x000A23D0, Port5: 0x000A2BD0,
Offset Formula: 0x000A03D0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:2	Reserved	RSVD 0x1	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1697: Analog Control 4 Register (n=0–6)
Offset: Port0: 0x000A03D4, Port1: 0x000A0BD4, Port2: 0x000A13D4, Port3: 0x000A1BD4, Port4: 0x000A23D4, Port5: 0x000A2BD4,
Offset Formula: 0x000A03D4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:0	Reserved	RSVD 0x5010	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1698: TX Training Control 25 Register (n=0–6)

Offset: Port0: 0x000A03D8, Port1: 0x000A0BD8, Port2: 0x000A13D8, Port3: 0x000A1BD8, Port4: 0x000A23D8, Port5: 0x000A2BD8,

Offset Formula: 0x000A03D8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:0	Reserved	RSVD 0x20	Reserved

Table 1699: TX Default 5 Register (n=0–6)

Offset: Port0: 0x000A03DC, Port1: 0x000A0BDC, Port2: 0x000A13DC, Port3: 0x000A1BDC, Port4: 0x000A23DC, Port5: 0x000A2BDC,

Offset Formula: 0x000A03DC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1600	Reserved

Table 1700: Clk150m Control 3 Register (n=0–6)

Offset: Port0: 0x000A03E0, Port1: 0x000A0BE0, Port2: 0x000A13E0, Port3: 0x000A1BE0, Port4: 0x000A23E0, Port5: 0x000A2BE0,

Offset Formula: 0x000A03E0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	PLL150M_DUTYCYCLE_EN	RW 0x0	1'b0 Save power and output clock have poor duty cycle (about 70%) 1'b1 Output clock with good duty cycle
14	PLL150M_FREERUN_EN	RW 0x0	1'b0: FLL loop close, loop lock state 1 = FLL loop open, freerun state
13:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:0	PLL150M_RSVD_IN[7:0]	RW 0x0	Reserved register bits

Table 1701: Signal Detection Calibration Register (n=0–6)

Offset: Port0: 0x000A03E4, Port1: 0x000A0BE4, Port2: 0x000A13E4, Port3: 0x000A1BE4, Port4: 0x000A23E4, Port5: 0x000A2BE4,

Offset Formula: 0x000A03E4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x6600	Reserved

Copyright © 2015 Marvell

Table 1702: Analog Control 5 Register (n=0–6)

Offset: Port0: 0x000A03E8, Port1: 0x000A0BE8, Port2: 0x000A13E8, Port3: 0x000A1BE8, Port4: 0x000A23E8, Port5: 0x000A2BE8,

Offset Formula: 0x000A03E8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x3AC	Reserved
4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3	Reserved	RSVD 0x0	Reserved
2:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1703: TX Default 7 Register (n=0–6)

Offset: Port0: 0x000A03EC, Port1: 0x000A0BEC, Port2: 0x000A13EC, Port3: 0x000A1BEC, Port4: 0x000A23EC, Port5: 0x000A2BEC,

Offset Formula: 0x000A03EC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1204	Reserved

Table 1704: TRX Training Control 9 Register (n=0–6)

Offset: Port0: 0x000A03F0, Port1: 0x000A0BF0, Port2: 0x000A13F0, Port3: 0x000A1BF0, Port4: 0x000A23F0, Port5: 0x000A2BF0,

Offset Formula: 0x000A03F0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x0	Reserved
8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:6	Reserved	RSVD 0x0	Reserved
5:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1705: TX Default 6 Register (n=0–6)

Offset: Port0: 0x000A03F4, Port1: 0x000A0BF4, Port2: 0x000A13F4, Port3: 0x000A1BF4, Port4: 0x000A23F4, Port5: 0x000A2BF4,

Offset Formula: 0x000A03F4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1303	Reserved

Table 1706: Squelch OOB Register (n=0–6)

Offset: Port0: 0x000A03F8, Port1: 0x000A0BF8, Port2: 0x000A13F8, Port3: 0x000A1BF8, Port4: 0x000A23F8, Port5: 0x000A2BF8,

Offset Formula: 0x000A03F8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:8	SQ_PKDETI_NON_OOB[1:0]	RW 0x2	Squech peak detector current setting for lower bandwith Refer to sq_oob_sel description
7:6	SQ_PKDETI_OOB[1:0]	RW 0x3	Squech peak detector current setting for higher bandwith This is used when oob is used
5:4	SQ_PKDETCAP_NON_OOB[1:0]	RW 0x1	Squech peak detector capacitor setting for lower bandwith This is used when oob is not used
3:2	SQ_PKDETCAP_OOB[1:0]	RW 0x0	Squech peak detector capacitor setting for higher bandwith This is used when oob is used
1	SQ_OOB_SEL	RW 0x0	Squech detection for OOB detection This is only used when sq_oob_sel_fm_reg or phy_isolation_mode is set to high 0 = use sq_pkdeti_non_oob, sq_pkdetcap_non_oob, sq_lpf_en_non_oob 1 = use sq_pkdeti_oob, sq_pkdetcap_oob, sq_lpf_en_oob
0	SQ_OOB_SEL_FM_REG	RW 0x0	sq_oob_sel control from register 0 = sq_oob_sel from pin, PIN_SQ_OOB_SEL 1 = sq_oob_sel from register, sq_oob_sel

Table 1707: Squelch Detect Calibration and VDD Calibration Result Register (n=0–6)

Offset: Port0: 0x000A0400, Port1: 0x000A0C00, Port2: 0x000A1400, Port3: 0x000A1C00, Port4: 0x000A2400, Port5: 0x000A2C00,

Offset Formula: 0x000A0400+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:8	Reserved	RSVD 0x0	Reserved
7:5	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1707: Squelch Detect Calibration and VDD Calibration Result Register (n=0–6) (Continued)
Offset: Port0: 0x000A0400, Port1: 0x000A0C00, Port2: 0x000A1400, Port3: 0x000A1C00, Port4: 0x000A2400, Port5: 0x000A2C00,
Offset Formula: 0x000A0400+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
4:0	Reserved	RSVD 0x0	Reserved

Table 1708: Align 90 Calibration Result Register (n=0–6)
Offset: Port0: 0x000A0404, Port1: 0x000A0C04, Port2: 0x000A1404, Port3: 0x000A1C04, Port4: 0x000A2404, Port5: 0x000A2C04,
Offset Formula: 0x000A0404+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1709: DCC Calibration Result Register (n=0–6)
Offset: Port0: 0x000A0408, Port1: 0x000A0C08, Port2: 0x000A1408, Port3: 0x000A1C08, Port4: 0x000A2408, Port5: 0x000A2C08,
Offset Formula: 0x000A0408+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:2	Reserved	RSVD 0x0	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1710: Calibration Result Register (n=0–6)
Offset: Port0: 0x000A040C, Port1: 0x000A0C0C, Port2: 0x000A140C, Port3: 0x000A1C0C, Port4: 0x000A240C, Port5: 0x000A2C0C,
Offset Formula: 0x000A040C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1711: RX DCC Calibration Result Register (n=0–6)
Offset: Port0: 0x000A0410, Port1: 0x000A0C10, Port2: 0x000A1410, Port3: 0x000A1C10, Port4: 0x000A2410, Port5: 0x000A2C10,
Offset Formula: 0x000A0410+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1711: RX DCC Calibration Result Register (n=0–6) (Continued)
Offset: Port0: 0x000A0410, Port1: 0x000A0C10, Port2: 0x000A1410, Port3: 0x000A1C10, Port4: 0x000A2410, Port5: 0x000A2C10,
Offset Formula: 0x000A0410+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x0	Reserved

Table 1712: Sampler Calibration Result Register (n=0–6)
Offset: Port0: 0x000A0414, Port1: 0x000A0C14, Port2: 0x000A1414, Port3: 0x000A1C14, Port4: 0x000A2414, Port5: 0x000A2C14,
Offset Formula: 0x000A0414+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Table 1713: CDR Control Register (n=0–6)
Offset: Port0: 0x000A0418, Port1: 0x000A0C18, Port2: 0x000A1418, Port3: 0x000A1C18, Port4: 0x000A2418, Port5: 0x000A2C18,
Offset Formula: 0x000A0418+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	RX_MAX_DFE_ADAPT_0[1:0]	RW 0x0	Number of dfe adaptations average in rx training when register TX_TRAIN_ENABLE and PIN_TX_TRAIN_ENABLE are both 0 0 = 1 1 = 2 2 = 4 3 = 8
13:12	RX_MAX_DFE_ADAPT_1[1:0]	RW 0x3	Number of dfe adaptations average in rx training when either register TX_TRAIN_ENABLE or PIN_TX_TRAIN_ENABLE is 1 0 = 1 1 = 2 2 = 4 3 = 8

Copyright © 2015 Marvell

Table 1713: CDR Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A0418, Port1: 0x000A0C18, Port2: 0x000A1418, Port3: 0x000A1C18, Port4: 0x000A2418, Port5: 0x000A2C18,

Offset Formula: 0x000A0418+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
11:9	CDR_MAX_DFE_ADAPT_0[2:0]	RW 0x1	Number of dfe adaptations average in phase training when register TX_TRAIN_ENABLE and PIN_TX_TRAIN_ENABLE are both 0 Purpose is to average out dfe variation. 0 = 1 1 = 2 2 = 4 3 = 8 4 = 16 5 = 32 6 = 64 7 = 128
8:6	CDR_MAX_DFE_ADAPT_1[2:0]	RW 0x4	Number of dfe adaptations average in phase training when either register TX_TRAIN_ENABLE or PIN_TX_TRAIN_ENABLE is 1 Purpose is to average out dfe variation. 0 = 1 1 = 2 2 = 4 3 = 8 4 = 16 5 = 32 6 = 64 7 = 128
5:0	Reserved	RSVD 0x0	Reserved

Table 1714: CDR Control Register (n=0–6)

Offset: Port0: 0x000A041C, Port1: 0x000A0C1C, Port2: 0x000A141C, Port3: 0x000A1C1C, Port4: 0x000A241C, Port5: 0x000A2C1C,

Offset Formula: 0x000A041C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x8032	Reserved

Table 1715: CDR Control Register (n=0–6)

Offset: Port0: 0x000A0420, Port1: 0x000A0C20, Port2: 0x000A1420, Port3: 0x000A1C20, Port4: 0x000A2420, Port5: 0x000A2C20,

Offset Formula: 0x000A0420+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:2	Reserved	RSVD 0x2008	Reserved
1	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1715: CDR Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A0420, Port1: 0x000A0C20, Port2: 0x000A1420, Port3: 0x000A1C20, Port4: 0x000A2420, Port5: 0x000A2C20,

Offset Formula: 0x000A0420+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
0	Reserved	RSVD 0x0	Reserved

Table 1716: CDR Control Register (n=0–6)

Offset: Port0: 0x000A0424, Port1: 0x000A0C24, Port2: 0x000A1424, Port3: 0x000A1C24, Port4: 0x000A2424, Port5: 0x000A2C24,

Offset Formula: 0x000A0424+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x25	Reserved

Table 1717: RX Train Midpoint Register (n=0–6)

Offset: Port0: 0x000A0428, Port1: 0x000A0C28, Port2: 0x000A1428, Port3: 0x000A1C28, Port4: 0x000A2428, Port5: 0x000A2C28,

Offset Formula: 0x000A0428+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0xFEE8	Reserved

Table 1718: RX Train Midpoint Register (n=0–6)

Offset: Port0: 0x000A042C, Port1: 0x000A0C2C, Port2: 0x000A142C, Port3: 0x000A1C2C, Port4: 0x000A242C, Port5: 0x000A2C2C,

Offset Formula: 0x000A042C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x3FB8	Reserved
1:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1719: TX Training Control 9 Register (n=0–6)

Offset: Port0: 0x000A0430, Port1: 0x000A0C30, Port2: 0x000A1430, Port3: 0x000A1C30, Port4: 0x000A2430, Port5: 0x000A2C30,

Offset Formula: 0x000A0430+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x8888	Reserved

Copyright © 2015 Marvell

Table 1720: TX Training Control 10 Register (n=0–6)

Offset: Port0: 0x000A0434, Port1: 0x000A0C34, Port2: 0x000A1434, Port3: 0x000A1C34, Port4: 0x000A2434, Port5: 0x000A2C34,

Offset Formula: 0x000A0434+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x8965	Reserved

Table 1721: TX Training Control 11 Register (n=0–6)

Offset: Port0: 0x000A0438, Port1: 0x000A0C38, Port2: 0x000A1438, Port3: 0x000A1C38, Port4: 0x000A2438, Port5: 0x000A2C38,

Offset Formula: 0x000A0438+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	Reserved	RSVD 0x2	Reserved
13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x1ECB	Reserved

Table 1722: TX Training Control 12 Register (n=0–6)

Offset: Port0: 0x000A043C, Port1: 0x000A0C3C, Port2: 0x000A143C, Port3: 0x000A1C3C, Port4: 0x000A243C, Port5: 0x000A2C3C,

Offset Formula: 0x000A043C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x801C	Reserved

Table 1723: G1 Setting 3 Register (n=0–6)

Offset: Port0: 0x000A0440, Port1: 0x000A0C40, Port2: 0x000A1440, Port3: 0x000A1C40, Port4: 0x000A2440, Port5: 0x000A2C40,

Offset Formula: 0x000A0440+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	Reserved	RSVD 0x0	Reserved
9	G1_FBCK_SEL	RW 0x0	Change the input of FBDIV FBCK_SEL=1 TXINTP FBCK_SEL=0 VCO

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1723: G1 Setting 3 Register (n=0–6) (Continued)

Offset: Port0: 0x000A0440, Port1: 0x000A0C40, Port2: 0x000A1440, Port3: 0x000A1C40, Port4: 0x000A2440, Port5: 0x000A2C40,

Offset Formula: 0x000A0440+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
8	G1_DFE_FREEZE_AFTER_TRAIN	RW 0x0	Freeze DFE after TX/RX training 0 = Do not freeze DFE taps after TX/RX training 1 = freeze DFE taps after TX/RX training
7	G1_FFE_SETTING_FORCE	RW 0x1	Feed Forward Equalization (FFE) Setting Force. Bypass the FFE table settings and use the FFE settings directly from register controls FFE_RES_SEL and FFE_CAP_SEL.
6:4	G1_FFE_RES_SEL[2:0]	RW 0x3	Feed Forward Equalization (FFE) Resistor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE. This field only works when ffe_setting_force is set to high.
3:0	G1_FFE_CAP_SEL[3:0]	RW 0xE	Feed Forward Equalization (FFE) Capacitor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE.

Table 1724: G1 Setting 4 Register (n=0–6)

Offset: Port0: 0x000A0444, Port1: 0x000A0C44, Port2: 0x000A1444, Port3: 0x000A1C44, Port4: 0x000A2444, Port5: 0x000A2C44,

Offset Formula: 0x000A0444+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:10	Reserved	RSVD 0x9	Reserved
9:8	G1_DFE_RES[1:0]	RW 0x1	Decision Feedback Equalization (DFE) Resolution. This setting affects the resolution of all DFE taps F0 through F5 0 = 3 mV 1 = 4 mV 2 = 5 mV 3 = 6 mV
7:3	G1_SQ_THRESH_RATIO[4:0]	RW 0x0	Squelch Detector Threshold ratio sq_thresh = sq_thresh_cal_result * ratio when SQ_THRESH_RATION != 0: ratio = SQ_THRESH_RATIO / 32 when SQ_THRESH_RATION == 0: ratio = 1
2:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1725: G2 Setting 3 Register (n=0–6)

Offset: Port0: 0x000A0448, Port1: 0x000A0C48, Port2: 0x000A1448, Port3: 0x000A1C48, Port4: 0x000A2448, Port5: 0x000A2C48,

Offset Formula: 0x000A0448+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	Reserved	RSVD 0x0	Reserved
9	G2_FBCK_SEL	RW 0x0	Change the input of FBDIV FBCK_SEL=1 TXINTP FBCK_SEL=0 VCO
8	G2_DFE_FREEZE_AFTER_TRAIN	RW 0x0	Freeze DFE after TX/RX training 0 = Do not freeze DFE taps after TX/RX training 1 = freeze DFE taps after TX/RX training
7	G2_FFE_SETTING_FORCE	RW 0x1	Feed Forward Equalization (FFE) Setting Force. Bypass the FFE table settings and use the FFE settings directly from register controls FFE_RES_SEL and FFE_CAP_SEL.
6:4	G2_FFE_RES_SEL[2:0]	RW 0x3	Feed Forward Equalization (FFE) Resistor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE. This field only works when ffe_setting_force is set to high.
3:0	G2_FFE_CAP_SEL[3:0]	RW 0xE	Feed Forward Equalization (FFE) Capacitor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE.

Table 1726: G2 Setting 4 Register (n=0–6)

Offset: Port0: 0x000A044C, Port1: 0x000A0C4C, Port2: 0x000A144C, Port3: 0x000A1C4C, Port4: 0x000A244C, Port5: 0x000A2C4C,

Offset Formula: 0x000A044C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	Reserved	RSVD 0x12	Reserved
9:8	G2_DFE_RES[1:0]	RW 0x1	Decision Feedback Equalization (DFE) Resolution. This setting affects the resolution of all DFE taps F0 through F5 0 = 3 mV 1 = 4 mV 2 = 5 mV 3 = 6 mV

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1726: G2 Setting 4 Register (n=0–6) (Continued)

Offset: Port0: 0x000A044C, Port1: 0x000A0C4C, Port2: 0x000A144C, Port3: 0x000A1C4C, Port4: 0x000A244C, Port5: 0x000A2C4C,

Offset Formula: 0x000A044C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
7:3	G2_SQ_THRESH_RATIO[4:0]	RW 0x0	Squelch Detector Threshold ratio sq_thresh = sq_thresh_cal_result * ratio when SQ_THRESH_RATION != 0: ratio = SQ_THRESH_RATIO / 32 when SQ_THRESH_RATION == 0: ratio = 1
2:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1727: G3 Setting 3 Register (n=0–6)

Offset: Port0: 0x000A0450, Port1: 0x000A0C50, Port2: 0x000A1450, Port3: 0x000A1C50, Port4: 0x000A2450, Port5: 0x000A2C50,

Offset Formula: 0x000A0450+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	Reserved	RSVD 0x0	Reserved
9	G3_FBCK_SEL	RW 0x0	Change the input of FBDIV FBCK_SEL=1 TXINTP FBCK_SEL=0 VCO
8	G3_DFE_FREEZE_AFTER_TRAIN	RW 0x0	Freeze DFE after TX/RX training 0 = Do not freeze DFE taps after TX/RX training 1 = freeze DFE taps after TX/RX training
7	G3_FFE_SETTING_FORCE	RW 0x0	Feed Forward Equalization (FFE) Setting Force. Bypass the FFE table settings and use the FFE settings directly from register controls FFE_RES_SEL and FFE_CAP_SEL.
6:4	G3_FFE_RES_SEL[2:0]	RW 0x3	Feed Forward Equalization (FFE) Resistor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE. This field only works when ffe_setting_force is set to high.
3:0	G3_FFE_CAP_SEL[3:0]	RW 0xE	Feed Forward Equalization (FFE) Capacitor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE.

Copyright © 2015 Marvell

Table 1728: G3 Setting 4 Register (n=0–6)

Offset: Port0: 0x000A0454, Port1: 0x000A0C54, Port2: 0x000A1454, Port3: 0x000A1C54, Port4: 0x000A2454, Port5: 0x000A2C54,

Offset Formula: 0x000A0454+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	Reserved	RSVD 0x24	Reserved
9:8	G3_DFE_RES[1:0]	RW 0x1	Decision Feedback Equalization (DFE) Resolution. This setting affects the resolution of all DFE taps F0 through F5 0 = 3 mV 1 = 4 mV 2 = 5 mV 3 = 6 mV
7:3	G3_SQ_THRESH_RATIO[4:0]	RW 0x0	Squelch Detector Threshold ratio sq_thresh = sq_thresh_cal_result * ratio when SQ_THRESH_RATION != 0: ratio = SQ_THRESH_RATIO / 32 when SQ_THRESH_RATION == 0: ratio = 1
2:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1729: G4_setting_3 Register (n=0–6)

Offset: Port0: 0x000A0458, Port1: 0x000A0C58, Port2: 0x000A1458, Port3: 0x000A1C58, Port4: 0x000A2458, Port5: 0x000A2C58,

Offset Formula: 0x000A0458+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:10	Reserved	RSVD 0x0	Reserved
9	G4_FBCK_SEL	RW 0x0	Change the input of FBDIV FBCK_SEL=1 TXINTP FBCK_SEL=0 VCO
8	G4_DFE_FREEZE_AFTER_TRAIN	RW 0x0	Freeze DFE after TX/RX training 0 = Do not freeze DFE taps after TX/RX training 1 = freeze DFE taps after TX/RX training
7	G4_FFE_SETTING_FORCE	RW 0x0	Feed Forward Equalization (FFE) Setting Force. Bypass the FFE table settings and use the FFE settings directly from register controls FFE_RES_SEL and FFE_CAP_SEL.
6:4	G4_FFE_RES_SEL[2:0]	RW 0x3	Feed Forward Equalization (FFE) Resistor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE. This field only works when ffe_setting_force is set to high.

Copyright © 2015 Marvell

Table 1729: G4_setting_3 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0458, Port1: 0x000A0C58, Port2: 0x000A1458, Port3: 0x000A1C58, Port4: 0x000A2458, Port5: 0x000A2C58,
Offset Formula: 0x000A0458+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
3:0	G4_FFE_CAP_SEL[3:0]	RW 0xE	Feed Forward Equalization (FFE) Capacitor Select. The combination of the FFE_RES_SEL and FFE_CAP_SEL affects the peak-to-DC gain as well as the fine adjust of the 3 dB bandwidth of the FFE.

Table 1730: G4 Setting 4 Register (n=0–6)
Offset: Port0: 0x000A045C, Port1: 0x000A0C5C, Port2: 0x000A145C, Port3: 0x000A1C5C, Port4: 0x000A245C, Port5: 0x000A2C5C,
Offset Formula: 0x000A045C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:10	Reserved	RSVD 0x3F	Reserved
9:8	G4_DFE_RES[1:0]	RW 0x1	Decision Feedback Equalization (DFE) Resolution. This setting affects the resolution of all DFE taps F0 through F5 0 = 3 mV 1 = 4 mV 2 = 5 mV 3 = 6 mV
7:3	G4_SQ_THRESH_RATIO[4:0]	RW 0x0	Squelch Detector Threshold ratio sq_thresh = sq_thresh_cal_result * ratio when SQ_THRESH_RATION != 0: ratio = SQ_THRESH_RATIO / 32 when SQ_THRESH_RATION == 0: ratio = 1
2:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1731: PCIe TX Preset Table 0 Register (n=0–6)
Offset: Port0: 0x000A0460, Port1: 0x000A0C60, Port2: 0x000A1460, Port3: 0x000A1C60, Port4: 0x000A2460, Port5: 0x000A2C60,
Offset Formula: 0x000A0460+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0xA765	Reserved

Copyright © 2015 Marvell

Table 1732: PCIe TX Preset Table 1 Register (n=0–6)

Offset: Port0: 0x000A0464, Port1: 0x000A0C64, Port2: 0x000A1464, Port3: 0x000A1C64, Port4: 0x000A2464, Port5: 0x000A2C64,

Offset Formula: 0x000A0464+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1234	Reserved

Table 1733: PCIe TX Preset Index Register (n=0–6)

Offset: Port0: 0x000A0468, Port1: 0x000A0C68, Port2: 0x000A1468, Port3: 0x000A1C68, Port4: 0x000A2468, Port5: 0x000A2C68,

Offset Formula: 0x000A0468+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:6	Reserved	RSVD 0x0	Reserved
5:4	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
3:0	TX_PRESET_INDEX[3:0]	RW 0x5	TX Preset Index During TX training, local PHY send reset command to remote PHY. There are four bits for reset signal. This register is used to determine which index will be used for reset command. 1 = Initialize coefficient 1 = P0, preshoot: 0dB, de-emphasis: -6dB 1 = reference 1 2 = P1, preshoot: 0dB, de-emphasis: -3.5dB 2 = Preset coefficient 2 = reference 2 3 = P2, preshoot: 0dB, de-emphasis: -4.4dB 3 = no equalization 4 = P3, preshoot, 0dB, de-emphasis: -2.5dB 5 = P4, preshoot, 0dB, de-emphasis: 0dB 6 = P5, preshoot, 1.9dB, de-emphasis: 0dB 7 = P6, preshoot, 2.5dB, de-emphasis: 0dB 8 = P7, preshoot, 3.5dB, de-emphasis: -6dB 9 = P8, preshoot, 3.5dB, de-emphasis: -3.5dB 10 = P9, preshoot, 3.5dB, de-emphasis: 0dB 11 = P10, preshoot, 0dB, de-emphasis: -9dB

Table 1734: TX Power Protection Control Register (n=0–6)

Offset: Port0: 0x000A046C, Port1: 0x000A0C6C, Port2: 0x000A146C, Port3: 0x000A1C6C, Port4: 0x000A246C, Port5: 0x000A2C6C,

Offset Formula: 0x000A046C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:15	Reserved	RSVD 0x1	Reserved

Copyright © 2015 Marvell

Table 1734: TX Power Protection Control Register (n=0–6) (Continued)

Offset: Port0: 0x000A046C, Port1: 0x000A0C6C, Port2: 0x000A146C, Port3: 0x000A1C6C, Port4: 0x000A246C, Port5: 0x000A2C6C,

Offset Formula: 0x000A046C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
14	TX_POWER_PROTECT_EN	RW 0x1	TX power protection enable If this bit is set to high, when local PHY receives coefficient request with single argument, if TX power reaches maximum, this coefficient will not be adjusted and return max/min status.
13:8	TX_POWER_MAX[5:0]	RW 0x1C	TX maximum power
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x5C	Reserved

Table 1735: DFE Control 27 Register (n=0–6)

Offset: Port0: 0x000A0470, Port1: 0x000A0C70, Port2: 0x000A1470, Port3: 0x000A1C70, Port4: 0x000A2470, Port5: 0x000A2C70,

Offset Formula: 0x000A0470+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	TX_MAX_DFE_ADAPT[1:0]	RW 0x3	Number of dfe adaptations average in tx training 0 = 1 1 = 2 2 = 4 3 = 8
13:6	Reserved	RSVD 0x0	Reserved
5:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1736: RX Training for Over Boost Register (n=0–6)

Offset: Port0: 0x000A0474, Port1: 0x000A0C74, Port2: 0x000A1474, Port3: 0x000A1C74, Port4: 0x000A2474, Port5: 0x000A2C74,

Offset Formula: 0x000A0474+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
8:0	Reserved	RSVD 0x101	Reserved

Copyright © 2015 Marvell

Table 1737: TX Training for Over Boost Register (n=0–6)

Offset: Port0: 0x000A0478, Port1: 0x000A0C78, Port2: 0x000A1478, Port3: 0x000A1C78, Port4: 0x000A2478, Port5: 0x000A2C78,

Offset Formula: 0x000A0478+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x1F1C	Reserved

Table 1738: RX Training Mode 0 Register (n=0–6)

Offset: Port0: 0x000A047C, Port1: 0x000A0C7C, Port2: 0x000A147C, Port3: 0x000A1C7C, Port4: 0x000A247C, Port5: 0x000A2C7C,

Offset Formula: 0x000A047C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0xF	Reserved
11:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1739: PCIe RX Preset Hint 1 Register (n=0–6)

Offset: Port0: 0x000A0480, Port1: 0x000A0C80, Port2: 0x000A1480, Port3: 0x000A1C80, Port4: 0x000A2480, Port5: 0x000A2C80,

Offset Formula: 0x000A0480+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x4F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x4F	Reserved

Table 1740: PCIe RX Preset Hint 2 Register (n=0–6)

Offset: Port0: 0x000A0484, Port1: 0x000A0C84, Port2: 0x000A1484, Port3: 0x000A1C84, Port4: 0x000A2484, Port5: 0x000A2C84,

Offset Formula: 0x000A0484+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1740: PCIe RX Preset Hint 2 Register (n=0–6) (Continued)
Offset: Port0: 0x000A0484, Port1: 0x000A0C84, Port2: 0x000A1484, Port3: 0x000A1C84, Port4: 0x000A2484, Port5: 0x000A2C84,
Offset Formula: 0x000A0484+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
14:8	Reserved	RSVD 0x4F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x4F	Reserved

Table 1741: PCIe RX Preset Hint 3 Register (n=0–6)
Offset: Port0: 0x000A0488, Port1: 0x000A0C88, Port2: 0x000A1488, Port3: 0x000A1C88, Port4: 0x000A2488, Port5: 0x000A2C88,
Offset Formula: 0x000A0488+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x4F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x4F	Reserved

Table 1742: PCIe RX Preset Hint 4 Register (n=0–6)
Offset: Port0: 0x000A048C, Port1: 0x000A0C8C, Port2: 0x000A148C, Port3: 0x000A1C8C, Port4: 0x000A248C, Port5: 0x000A2C8C,
Offset Formula: 0x000A048C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x4F	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x4F	Reserved

Copyright © 2015 Marvell

Table 1743: PCIe RX Preset Hint 5 Register (n=0–6)

Offset: Port0: 0x000A0490, Port1: 0x000A0C90, Port2: 0x000A1490, Port3: 0x000A1C90, Port4: 0x000A2490, Port5: 0x000A2C90,

Offset Formula: 0x000A0490+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x5555	Reserved

Table 1744: PCIe RX Preset Hint 6 Register (n=0–6)

Offset: Port0: 0x000A0494, Port1: 0x000A0C94, Port2: 0x000A1494, Port3: 0x000A1C94, Port4: 0x000A2494, Port5: 0x000A2C94,

Offset Formula: 0x000A0494+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x5555	Reserved

Table 1745: PCIe RX Preset Hint 7 Register (n=0–6)

Offset: Port0: 0x000A0498, Port1: 0x000A0C98, Port2: 0x000A1498, Port3: 0x000A1C98, Port4: 0x000A2498, Port5: 0x000A2C98,

Offset Formula: 0x000A0498+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:12	TX_VMA_MIN[3:0]	RW 0x4	TX mininum voltage modulation amplitude During TX training, tx_amp_index+tx_amp_offset-tx_emph0_index-tx_emph1_index must be more than or equal to tx_vma_min
11	TX_VMA_PROTECT_EN	RW 0x1	TX voltage modulation amplitude protection enable 0 = disable TX VMA protection 1 = enable TX VMA protection. . If tx_amp_index+tx_amp_offset-tx_emph0_index-tx_emph1_index <= tx_vma_min, the coefficient will be set to maximum for emphasis or minimum for main tap.
10:0	Reserved	RSVD 0xC0	Reserved

Table 1746: DFE Control 28 Register (n=0–6)

Offset: Port0: 0x000A049C, Port1: 0x000A0C9C, Port2: 0x000A149C, Port3: 0x000A1C9C, Port4: 0x000A249C, Port5: 0x000A2C9C,

Offset Formula: 0x000A049C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1746: DFE Control 28 Register (n=0–6) (Continued)
Offset: Port0: 0x000A049C, Port1: 0x000A0C9C, Port2: 0x000A149C, Port3: 0x000A1C9C, Port4: 0x000A249C, Port5: 0x000A2C9C,
Offset Formula: 0x000A049C+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
9:0	Reserved	RSVD 0x28	Reserved

Table 1747: F1_BUF_OS Adjust 0 Register (n=0–6)
Offset: Port0: 0x000A04A0, Port1: 0x000A0CA0, Port2: 0x000A14A0, Port3: 0x000A1CA0, Port4: 0x000A24A0, Port5: 0x000A2CA0,
Offset Formula: 0x000A04A0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x0	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Table 1748: F1_BUF_OS Adjust 1 Register (n=0–6)
Offset: Port0: 0x000A04A4, Port1: 0x000A0CA4, Port2: 0x000A14A4, Port3: 0x000A1CA4, Port4: 0x000A24A4, Port5: 0x000A2CA4,
Offset Formula: 0x000A04A4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x0	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1749: F1_BUF_OS Adjust 2 Register (n=0–6)

Offset: Port0: 0x000A04A8, Port1: 0x000A0CA8, Port2: 0x000A14A8, Port3: 0x000A1CA8, Port4: 0x000A24A8, Port5: 0x000A2CA8,

Offset Formula: 0x000A04A8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x0	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Table 1750: F1_BUF_OS Adjust 3 Register (n=0–6)

Offset: Port0: 0x000A04AC, Port1: 0x000A0CAC, Port2: 0x000A14AC, Port3: 0x000A1CAC, Port4: 0x000A24AC, Port5: 0x000A2CAC,

Offset Formula: 0x000A04AC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:8	Reserved	RSVD 0x0	Reserved
7	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
6:0	Reserved	RSVD 0x0	Reserved

Table 1751: TX Default 8 Register (n=0–6)

Offset: Port0: 0x000A04B0, Port1: 0x000A0CB0, Port2: 0x000A14B0, Port3: 0x000A1CB0, Port4: 0x000A24B0, Port5: 0x000A2CB0,

Offset Formula: 0x000A04B0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0xD63	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1752: TX Default 9 Register (n=0–6)

Offset: Port0: 0x000A04B4, Port1: 0x000A0CB4, Port2: 0x000A14B4, Port3: 0x000A1CB4, Port4: 0x000A24B4, Port5: 0x000A2CB4,

Offset Formula: 0x000A04B4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0xE44	Reserved

Table 1753: TX Default 10 Register (n=0–6)

Offset: Port0: 0x000A04B8, Port1: 0x000A0CB8, Port2: 0x000A14B8, Port3: 0x000A1CB8, Port4: 0x000A24B8, Port5: 0x000A2CB8,

Offset Formula: 0x000A04B8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x1105	Reserved

Table 1754: TX Default 11 Register (n=0–6)

Offset: Port0: 0x000A04BC, Port1: 0x000A0CBC, Port2: 0x000A14BC, Port3: 0x000A1CBC, Port4: 0x000A24BC, Port5: 0x000A2CBC,

Offset Formula: 0x000A04BC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0xCA0	Reserved

Table 1755: DFE 30 Register (n=0–6)

Offset: Port0: 0x000A04C0, Port1: 0x000A0CC0, Port2: 0x000A14C0, Port3: 0x000A1CC0, Port4: 0x000A24C0, Port5: 0x000A2CC0,

Offset Formula: 0x000A04C0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0xC9F	Reserved
3:0	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1756: Internal Digital Reserved 0 Register (n=0–6)

Offset: Port0: 0x000A05EC, Port1: 0x000A0DEC, Port2: 0x000A15EC, Port3: 0x000A1DEC, Port4: 0x000A25EC, Port5: 0x000A2DEC,

Offset Formula: 0x000A05EC+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0xF0	Reserved

Table 1757: Analog Reserved 0 Register (n=0–6)

Offset: Port0: 0x000A05F0, Port1: 0x000A0DF0, Port2: 0x000A15F0, Port3: 0x000A1DF0, Port4: 0x000A25F0, Port5: 0x000A2DF0,

Offset Formula: 0x000A05F0+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1758: Internal CID 0 Register (n=0–6)

Offset: Port0: 0x000A05F4, Port1: 0x000A0DF4, Port2: 0x000A15F4, Port3: 0x000A1DF4, Port4: 0x000A25F4, Port5: 0x000A2DF4,

Offset Formula: 0x000A05F4+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x3211	Reserved

Table 1759: Internal CID 1 Register (n=0–6)

Offset: Port0: 0x000A05F8, Port1: 0x000A0DF8, Port2: 0x000A15F8, Port3: 0x000A1DF8, Port4: 0x000A25F8, Port5: 0x000A2DF8,

Offset Formula: 0x000A05F8+0x800*n: where n (0-6) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x211	Reserved

Table 1760: Lane Configuration 0 Register (n=0–1)

Offset: Port0: 0x000A0600, Port1: 0x000A0E00

Offset Formula: 0x000A0600+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1760: Lane Configuration 0 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0600, Port1: 0x000A0E00
Offset Formula: 0x000A0600+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	CFG_FORCE_RXPRESENT[1:0]	RW 0x0	Receiver detection override 0 = use value from UNIPHY 1 = force receiver NOT detected for this lane 2 = force receiver detected for this lane 3 = force receiver detected for this lane
13	CFG_FAST_SYNCH	RW 0x0	Fast synchronization at 125 MHz pclk frequency 0 = single-cycle synchronization 1 = half-cycle synchronization
12:9	CFG_ELB_THRESHOLD[3:0]	RW 0x8	Elastic buffer quiescent threshold 5 = minimum value for common REFCLK 8 = 600 ppm adjustment 9 = beyond 600 ppm adjustment 10 = VA: maximum value for 600 ppm adjustment 13 = VD: maximum value for common REFCLK
8:5	CFG_TX_ALIGN_POS[3:0]	RW 0x0	TX alignment shift position (for dphy_ana_txdata) 0 = shift by 0 bits 1 = shift by 1 bits 2 = shift by 1 bits 3 = shift by 2 bits 4 = shift by 2 bits 5 = shift by 3 bits 6 = shift by 3 bits 7 = shift by 4 bits 8 = shift by 5 bits 9 = shift by 6 bits 10 = VA: shift by 6 bits 11 = VB: shift by 7 bits 12 = VC: shift by 7 bits 13 = VD: shift by 8 bits 14 = VE: shift by 8 bits 15 = VF: shift by 9 bits
4	PRD_TXSWING	RW 0x0	replaces mac_phy_txswing when mode_margin_override=1
3:1	PRD_TXMARGIN[2:0]	RW 0x0	replaces mac_phy_txmargin when mode_margin_override=1
0	PRD_TXDEEMPH0	RW 0x0	replaces mac_phy_txdeemph[0] when mode_margin_override=1

Copyright © 2015 Marvell

Table 1761: Lane Configuration 1 Register (n=0–1)
Offset: Port0: 0x000A0604, Port1: 0x000A0E04
Offset Formula: 0x000A0604+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15	PRD_TXDEEMPH1	RW 0x0	replaces mac_phy_txdeemph[1] when mode_margin_override=1
14	CFG_GEN2_DLY_RXDATA	RW 0x0	Delay RxData latency for one pclk cycle when read it out from rx fifo,take effects only for 8bit PIPE interface at GEN1 speed or 16b PIPE interface at GEN3 speed
13	CFG_BUNDLE_FIFO	RW 0x1	Bundle read pointers of all lanes at tx fifo and rx fifo ,take effects only for 8bit PIPE interface at GEN1 speed or 16b PIPE interface at GEN3 speed 0 = unbundled 1 = bundled read pointer of all lanes at txfifo and rxfifo while 32 pipe inf
12	CFG_GEN_DLY_TXDATA	RW 0x0	Delay TxData latency for one dpclk cycle when read it out from tx fifo (takes ettect only for 32-bit PIPE interface 0 = no delay 1 = Delay TxData latency for one dpclk cycle when read it out from tx fifo
11	CFG_USE_GEN3_PLL_CAL	RW 0x0	Use gen3 speed for PLL calibration 0 = use gen1 speed for PLL calibration 1 = use gen3 speed for PLL calibration
10	CFG_USE_GEN2_PLL_CAL	RW 0x0	Use gen2 speed for PLL calibration 0 = use gen1 speed for PLL calibration 1 = use gen2 speed for PLL calibration
9	CFG_USE_MAX_PLL_RATE	RW 0x0	Use max PLL rate mode at Common PHY. 0 = non-max PLL rate mode at Common PHY 1 = max PLL rate mode at Common PHY mode
8	CFG_SPD_CHANGE_WAIT	RW 0x0	Speed change wait period when max PLL rate mode. 0 = 3.2us 1 = 6.4us
7	CFG_DISABLE_TXDETVAL	RW 0x0	Disable txdetrx valid signal during TX Detect RX 0 = enable txdetrx valid signal for TX Detect RX 1 = disable txdetrx valid signal for TX Detect RX
6	CFG_TXDETRX_MODE	RW 0x1	Tx Detect Rx mode 0 = Tx Detect Rx at low-z mode 1 = Tx Detect Rx at high-z mode
5	CFG_ALIGN_IDLE_HIZ	RW 0x0	Align the phase of internal idle high-z control signal with idle low-z control signal 0 = not aligned high-z off before low-off 1 = aligned both high-z off and low-off at the same pahse

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1761: Lane Configuration 1 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0604, Port1: 0x000A0E04
Offset Formula: 0x000A0604+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
4:3	CFG_GEN2_TXDATA_DLY[1:0]	RW 0x2	Tx datapath delayed latency (to accommodate high-z off latency of internal analog circuit Gen2 mode 0 = no delay 1 = delayed by 1 clock cycle 2 = delayed by 2 clock cycles 3 = delayed by 3 clock cycles
2:1	CFG_GEN1_TXDATA_DLY[1:0]	RW 0x0	Tx datapath delayed latency (to accommodate high-z off latency of internal analog circuit Gen1 mode 0 = no delay 1 = delayed by 1 clock cycle 2 = delayed by 2 clock cycles 3 = delayed by 3 clock cycles
0	CFG_TXELECIDLE_MODE	RW 0x1	Tx Electrical Idle mode enable 0 = Transmitter is at low impedance mode during Electrical Idle 1 = Transmitter is at high impedance mode during Electrical Idle

Table 1762: Lane Status 0 Register (n=0–1)
Offset: Port0: 0x000A0608, Port1: 0x000A0E08
Offset Formula: 0x000A0608+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15	PM_TX_VCMHOLD_EN	RO 0x0	control comphy tx common mode 0 = disable comphy tx common mode holder 1 = enavle comphy tx common mode holder
14	PM_BEACON_RX_EN	RO 0x0	low- frequency beacon detection enable 0 = disable low-frequency beacon detection 1 = enable low-frequency beacon detection
13	PM_BEACON_TX_EN	RO 0x0	beacon transmission enable 0 = no operation 1 = trigger comphy to transmit beacon
12	PM_PU_IVREF	RO 0x0	power up comphy current and voltage reference 0 = power down 1 = power up
11	PM_TXDETECTRX_EN	RO 0x0	control comphy to perform tx receiver detection 0 = no operation 1 = triggers comphy to perform tx recever detection

Copyright © 2015 Marvell

Table 1762: Lane Status 0 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0608, Port1: 0x000A0E08
Offset Formula: 0x000A0608+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
10	PM_TX_IDLE_HIZ	RO 0x0	control comphy tx driver idle in high impedance mode 0 = tx driver at low impedance mode 1 = tx driver at high impedance mode
9	PM_TX_IDLE_LOZ	RO 0x0	Control comphy tx driver 0 = tx driver ouput valid 1 = tx driver at common mode voltage (idle)
8	PM_RX_INIT	RO 0x0	Control Comphy receiver initialization 0 = no opertation 1 = triggers the comphy to start acquistition for phase of DEF FFE
7:6	PM_RX_RATE_SEL[1:0]	RO 0x0	Control Comphy RX signaling rate 0 = use 2.5GT/s signal rate 1 = use 5.0GT/s signal rate 2 = use 8.0GT/s signal rate 3 = reserved
5:4	PM_TX_RATE_SEL[1:0]	RO 0x0	Control Comphy TX signaling rate 0 = use 2.5GT/s signal rate 1 = use 5.0GT/s signal rate 2 = use 8.0GT/s signal rate 3 = reserved
3	PM_PU_RX	RO 0x0	Power up Comphy Receiver 0 = power dwon 1 = power up
2	PM_PU_TX	RO 0x0	Power up Comphy Transmitter 0 = power dwon 1 = power up
1	PM_PU_PLL	RO 0x0	Power up Comphy PLL 0 = power dwon 1 = power up
0	PM_RESET	RO 0x0	Common PHY reset 0 = normal working mode 1 = reset mode

Copyright © 2015 Marvell

Table 1763: Lane Status 1 Register (n=0–1)
Offset: Port0: 0x000A060C, Port1: 0x000A0E0C
Offset Formula: 0x000A060C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:10	PM_STATE[5:0]	RO 0x0	PM state[5:0] 0 = Reset 2 = PHY calibration 13 = GEN1 L1 CLKREQ state 15 = P2 state (link L2) 17 = GEN2 L1 CLKREQ state 21 = TX receiver detection 24 = GEN1 P1 state 25 = GEN1 DATA PATH reset 26 = GEN1 p0 or p0s state 27 = GEN1 clock off for speed change 28 = GEN2 p1 state 29 = GEN2 data path reset 30 = GEN2 p0 or p0s state 31 = GEN2 clock off for speed change 37 = GEN3 L1 CLKREQ state 40 = GEN3 P1 state 41 = GEN3 data path reset 42 = GEN3 p0 or p0s state 43 = GEN3 clock off for speed change
9	PM_CLK_REQ_N	RO 0x0	CLKREQ control status 0 = request to enable REFCLK 1 = request to disable REFCLK
8	PM_RPCLK_125	RO 0x0	DPCLK is 125Mhz
7	PM_PIPE_32B	RO 0x0	PIPE is opertating in 32bit mode
6	PM_PIPE_8B	RO 0x0	PIPE is opertating in 8bit mode
5	PM_ASYNC_RST_N	RO 0x0	PIPE internal power management reset 0 = reset mode 1 = normal working mode
4	PM_DP_RST_N	RO 0x0	PIPE internal PCS data-path reset 0 = reset mode 1 = normal working mode
3	PM_OSCCLK_AUX_CLK_EN	RO 0x0	AUX_CLK switchinh between txdclk and oscclk status 0 = AUX_CLK source is txdclk from pll 1 = AUX_CLK source is OSCCLK from SOC
2	PM_OSCCLK_PCLK_EN	RO 0x0	PCLK switching between txdclk and oscclk status 0 = pclk source si txdclk from pll 1 = pclk source is oscclk from soc

Copyright © 2015 Marvell

Table 1763: Lane Status 1 Register (n=0–1) (Continued)
Offset: Port0: 0x000A060C, Port1: 0x000A0E0C
Offset Formula: 0x000A060C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
1	PM_PCLK_DPCLK_EN	RO 0x0	Pipe internal PCS data-path clock status 0 = dpclk is disabled 1 = dpclk is enabled
0	PM_TXDCLK_PCLK_EN	RO 0x0	PCLK output enable status: 0 = PCLK is disabled 1 = PCLK is enabled, indicates that the PHY is ready and send out PCLK to the MAC

Table 1764: Lane Status 2 Register (n=0–1)
Offset: Port0: 0x000A0610, Port1: 0x000A0E10
Offset Formula: 0x000A0610+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	MAC_PHY_TXDETECTRX_LOOPBACK	RO 0x0	Control TX receiver detection or loopback from MAC 0 = normal operation 1 = trigger tx receiver detection or rx to tx loopback
14	MAC_PHY_TXELECIDLE	RO 0x0	contrl tx electrical idle from MAC 0 = sending data or beacon 1 = set transmitter to electrical idle
13:12	MAC_PHY_POWERDOWN[1:0]	RO 0x0	control power state from MAC 0 = P0, normal operation 1 = P0s, power saving state, low recovery time latency 2 = P1, lower power state, longer recovery time latency 3 = P2, lowest power state, longest recovery time latency
11:10	MAC_PHY_RATE[1:0]	RO 0x0	control signaling Rate from MAC 0 = use 2.5GT/s signal rate 1 = use 5.0GT/s signal rate 2 = use 8.0GT/s signal rate 3 = reserved
9	PHY_MAC_PHYSTATUS	RO 0x0	PHY status at PIPE interface, used to communicate completion of PHY functions such as reset, power state transition, rate changes and receiver detection
8	PHY_MAC_RXVALID	RO 0x0	RX data valid status at PIPE interface 0 = no valid data received 1 = indicate symbol lock and valid data at PIPE interface

Copyright © 2015 Marvell

Table 1764: Lane Status 2 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0610, Port1: 0x000A0E10
Offset Formula: 0x000A0610+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
7	PHY_MAC_RXELECIDLE	RO 0x0	RX Electrical IDLE status at PIPE interface 0 = receiving data or beacon 1 = indicates detection of an electrical idle
6	ANA_DPHY_PLL_READY_TX	RO 0x0	Comphy PLL ready for TX 0 = comphy tx clock is not ready 1 = indicates that Comphy tx clock is ready and TX can sent out data
5	MAC_PHY_RX_TERMINATION	RO 0x0	Receiver termination control from MAC for USB3 mode
4	ANA_DPHY_PLL_READY_RX	RO 0x0	Comphy PLL ready for RX 0 = comphy rx clock is not ready 1 = indicates that Comphy RX clock is ready and RX can receive data
3	ANA_DPHY_TXDETRX_VALID	RO 0x0	Comphy TX detect RX ouput valid 0 = TX detect Rx output is not valid 1 = TX detect Rx output is valid for ana_dphy_rxpresent
2	ANA_DPHY_RX_INIT_DONE	RO 0x0	Comphy receiver initialization done status 0 = Comphy DTL and DFE is no optimized 1 = indicate that Comphy DTL and DFE have entered optimized states
1	ANA_DPHY_SQ_DETECTED	RO 0x0	Comphy squelch detector ouput status 0 = signal detected 1 = no signal detected
0	ANA_DPHY_RXPRESENT	RO 0x0	Tx receiver detection status: 0 = receiver is not present 1 = receiver is present (during Tx receiver detection

Table 1765: Lane Beacon Control Register (n=0–1)
Offset: Port0: 0x000A0614, Port1: 0x000A0E14
Offset Formula: 0x000A0614+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	BEACON_DETECTED	RO 0x0	Low-frequency beacon is detected
14	CFG_POWER_SETTLE_WAIT	RW 0x1	Wait for power settle before sending LFPS when transition into P2 or P3 state USB3 mode. 0 = Send LFPS as long as TxElecIdle is de-asserted 1 = Wait for power state settle before sending LFPS even TxElecIdle is de-asserted

Copyright © 2015 Marvell

Table 1765: Lane Beacon Control Register (n=0–1) (Continued)
Offset: Port0: 0x000A0614, Port1: 0x000A0E14
Offset Formula: 0x000A0614+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
13:8	CFG_RXEIDETECT_DLY[5:0]	RW 0xA	PIPE signal RxElecIdle turn on (after squelch detector was previously disabled) delay timer value. Counts at oscclk domain.
7	CFG_IVREF_MODE	RW 0x1	Serdes analog IVREF control mode during low power (PCIE: P1.CLKREQ and P2; USB3: P3) states 0 = PU_IVREF is 0 1 = PU_IVREF is 1
6	CFG_BEACON_MODE	RW 0x0	Beacon/LFPS protocol control mode 0 = new protocol 1 = legacy protocol
5:2	CFG_BEACON_TXLOZ_WAIT[3:0]	RW 0x0	Beacon transmit low-z wait period. In units of 20us
1	CFG_BEACON_RX_EN	RW 0x0	Rx beacon mode enable
0	CFG_BEACON_TX_EN	RW 0x0	Tx beacon mode enable

Table 1766: Lane Configuration 2 Register (n=0–1)
Offset: Port0: 0x000A0618, Port1: 0x000A0E18
Offset Formula: 0x000A0618+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13	CFG_BLK_ALIGN_CTRL[2]	RW 0x0	block alignment control 0 = reset block alignment state machine until detecting differential signaling 1 = reset block alignment state machine until Comphy is ready. Rx_init is done and/or rx train is completed
12:11	CFG_BLK_ALIGN_CTRL[1:0]	RW 0x0	Enable pin MAC_PHY_BLOCKALIGNCTRL on controlling block alignment logic 0 = disable pin MAC_PHY_BLOCKALIGNCTRL on controlling block alignment logic 1 = disable pin MAC_PHY_BLOCKALIGNCTRL on controlling block alignment logic 2 = when MAC_PHY_BLOCKALIGNCTRL = 0, disable searching new block alignment, when MAC_PHY_BLOCKALIGNCTRL = 1, enable searching new block alignment 3 = when MAC_PHY_BLOCKALIGNCTRL = 0, reset block alignment state machine, when MAC_PHY_BLOCKALIGNCTRL = 1, enable searching new block alignment
10	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information

Table 1766: Lane Configuration 2 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0618, Port1: 0x000A0E18
Offset Formula: 0x000A0618+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
9	CFG_RXDATA_SKIP_SEL	RW 0x0	Select every N 128b/130b blocks for the PHY to instruct the MAC to ignore RxData for one (or two) PCLK cycle at the Rx PIPE interface mode. (by asserting RxDataSkip, or de-asserting RxDataValid) while operating at Gen3 16-bit PIPE 0 = every 16 blocks to ignore RxData for two PCLK cycles 1 = every 8 blocks to ignore RxData for one PCLK cycle
8:7	CFG_GEN3_TXDATA_DLY[1:0]	RW 0x0	Tx datapath delayed latency (to accommodate high-z off latency of internal analog circuit Gen3 mode. 0 = no delay 1 = delayed by 1 clock cycle 2 = delayed by 2 clock cycle 3 = delayed by 3 clock cycle
6:5	CFG_GEN3_TXELECIDLE_DLY[1:0]	RW 0x0	ComPHY tx_idle_hiz Timing Gen3 0 = tx_idle_hiz is synchronous with txdata 1 = delayed by 1 clock cycle 2 = delayed by 2 clock cycle 3 = delayed by 3 clock cycle
4:0	Reserved	RSVD 0x1	Reserved

Table 1767: Lane Configuration 3 Register (n=0–1)
Offset: Port0: 0x000A061C, Port1: 0x000A0E1C
Offset Formula: 0x000A061C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0xE28	Reserved

Table 1768: Lane Configuration 4 Register (n=0–1)
Offset: Port0: 0x000A0620, Port1: 0x000A0E20
Offset Formula: 0x000A0620+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1768: Lane Configuration 4 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0620, Port1: 0x000A0E20
Offset Formula: 0x000A0620+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
12:8	CFG_REF_FREF_SEL[4:0]	RW 0x0	Reference Frequency Select (MHz) for usb3 mode 0 = 100 1 = 20 2 = 25 3 = 40 4 = 50 5 = 62.5 6 = 75 7 = 125 8 = 150 9 = 156.25 10 = 250 11 = 30 12 = V0C: 31.25 13 = 26 14 = V0E: 19.2 15 = V0F: 38.4
7	CFG_SSC_CTRL	RW 0x0	Spread Spectrum Clock Enable 0 = Disable SSC 1 = Enable SSC
6	CFG_DFE_OVERRIDE	RW 0x0	Override PHY DFE control pins 0 = normal operation 1 = override PIN_DFE_EN, PIN_DFE_PAT_DIS, and PIN_DFE_UPDATE_DIS by cfg_dfe_ctrl[2:0].
5	CFG_DFE_UPDATE_SEL	RW 0x0	PHY PIN_DFE_UPDATE_DIS pin control select when cfg_dfe_override = 0. 0 = enable DFE adaptation 1 = Disable DFE adaptation but keep DFE current coefficient values for PCIE mode, or controlled by inverted mac_phy_rxeqtraining pin for USB3 mode.
4	CFG_DFE_PAT_SEL	RW 0x1	PHY PIN_DFE_PAT_DIS pin control select when cfg_dfe_override = 0. 0 = Disable DFE Pattern Protection function 1 = Enable DFE Pattern Protection function for PCIE mode, or controlled by mac_phy_rxeqtraining pin for USB3 mode
3	CFG_DFE_EN_SEL	RW 0x0	PHY PIN_DEF_EN pin control selest when cfg_dfe_override = 0. DFE is always disabled at PCIe Gen1 speed, and is always enabled at PCIe Gen3 speed and USB3 mode 0 = DFE is disabled at PCIe Gen2 speed 1 = DFE is enabled at PCIe Gen2 speed
2:0	CFG_DFE_CTRL[2:0]	RW 0x3	Controls PHY DFE signals when cfg_dfe_override = 1. Bit[0]: PIN_DFE_EN Bit[1]: PIN_DFE_PAT_DIS Bit[2]: PIN_DFE_UPDATE_DIS

Copyright © 2015 Marvell

Table 1769: Lane Configuration 5 Register (n=0–1)
Offset: Port0: 0x000A0624, Port1: 0x000A0E24
Offset Formula: 0x000A0624+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:5	Reserved	RSVD 0x0	Reserved
4	CFG_RXEI_DG_WEIGHT	RW 0x0	RxEelcIdle de-glitching tap weight 0 = 3-tap de-glitching 1 = 1-tap de-glitching
3	CFG_RXEIDET_DG_EN	RW 0x0	enable rx electrical idle deglitch 0 = disable the RxElecIdle de-glitch logic 1 = enable the RxElecIdle de-glitch logic
2	CFG_RX_EQ_CTRL	RW 0x0	PHY Rx Training enable while in USB3 LTSSM Polling.RxEQ state 0 = disable Rx Training 1 = enable Rx Training
1	CFG_SQ_DET_SEL	RW 0x0	squelch detect signal select for rx_init control 0 = use squelch detect signal directly from serdes 1 = use filtered squelch detect signal
0	CFG_RX_INIT_SEL	RW 0x0	rx_init control select 0 = do not reset rx_init control state machine when detect LFPS signaling 1 = reset rx_init control state machine when detect LFPS signaling

Table 1770: Lane Status 3 Register (n=0–1)
Offset: Port0: 0x000A0628, Port1: 0x000A0E28
Offset Formula: 0x000A0628+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	MAC_PHY_TXCOMPLIANCE[1:0]	RO 0x0	TX compliance control form MAC
13	PM_REFCLK_VALID	RO 0x0	External REFCLK detection status 0 = REFCLK is not valid 1 = REFCLK is valid
12	ANA_REFCLK_DIS_ACK	RO 0x0	Comphy REFCLK disable ackledgement 0 = normal opertaion 1 = Comphy is ready to disable REFCLK
11	PM_REFCLK_DIS	RO 0x0	Control disabling REFCLK at Comphy 0 = normal opertaion 1 = request to disable REFCLK at Comphy

Copyright © 2015 Marvell

Table 1770: Lane Status 3 Register (n=0–1) (Continued)
Offset: Port0: 0x000A0628, Port1: 0x000A0E28
Offset Formula: 0x000A0628+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
10	PM_PU_SQ	RO 0x0	Power up COMPHY squelch detector 0 = power down 1 = power up
9:0	ANA_RESERVED_STATUS[9:0]	RO 0x0	reserved

Table 1771: Lane TX Common Mode Configuration Register (n=0–1)
Offset: Port0: 0x000A062C, Port1: 0x000A0E2C
Offset Formula: 0x000A062C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13	CFG_P0S_IDLE_HIZ_DIS	RW 0x1	Disable tx_idle_hiz signal during P0S state and speed change.
12	CFG_HIZ_CAL_TIMER_EN	RW 0x0	Hiz Calibration Timer enable.
11:8	CFG_HIZ_CAL_WAIT[3:0]	RW 0x4	Hiz Calibration wait timer
7	CFG_DELAY_P12_PHYST	RW 0x0	Delay P1 and P2 entry PhyStatus to accommodate Hiz Calibration time at ComPHY.
6	CFG_DELAY_TDR_PHYST	RW 0x0	Delay TxDetectRx PhyStatus to the MAC after receiving TxDetecrRx valid (ackledgement) from ComPHY
5:0	CFG_TXCMN_DIS_DLY[5:0]	RW 0x14	Tx Common Mode turn on (after previously disabled) delay timer value. Counts at oscclk domain.

Table 1772: USB Datapath Configuration 1 Register (n=0–1)
Offset: Port0: 0x000A0684, Port1: 0x000A0E84
Offset Formula: 0x000A0684+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14:0	Reserved	RSVD 0xCF8	Reserved

Copyright © 2015 Marvell

Table 1773: Lane Equalization Configuration 0 Register (n=0–1)
Offset: Port0: 0x000A069C, Port1: 0x000A0E9C
Offset Formula: 0x000A069C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	CFG_USE_CTRL_FLD_RST	RW 0x0	Select phy ctrl_field_reset command while performing remote tx training. 0 = do not use phy ctrl_field_reset command 1 = use phy ctrl_field_reset command
14:13	CFG_SEL_EQ_STATUS[1:0]	RW 0x0	Select internal equalization status signals to be read out at LANE_EQ_STATUS register.
12	CFG_PHY_RC_EP	RW 0x0	PHY operation mode select. 0 = Endpoint 1 = Root Complex
11:6	CFG_EQ_LF[5:0]	RW 0xC	Transmitter low frequency parameter (LF)
5:0	CFG_EQ_FS[5:0]	RW 0x22	Transmitter full swing parameter (FS)

Table 1774: Lane Equalization Configuration 1 Register (n=0–1)
Offset: Port0: 0x000A06A0, Port1: 0x000A0EA0
Offset Formula: 0x000A06A0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x4041	Reserved

Table 1775: Lane Equalization Configuration 2 Register (n=0–1)
Offset: Port0: 0x000A06A4, Port1: 0x000A0EA4
Offset Formula: 0x000A06A4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x201	Reserved

Table 1776: Lane Equalization Preset Configuration 0 Register (n=0–1)
Offset: Port0: 0x000A06A8, Port1: 0x000A0EA8
Offset Formula: 0x000A06A8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Copyright © 2015 Marvell

Table 1776: Lane Equalization Preset Configuration 0 Register (n=0–1) (Continued)
Offset: Port0: 0x000A06A8, Port1: 0x000A0EA8
Offset Formula: 0x000A06A8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
11:0	Reserved	RSVD 0x719	Reserved

Table 1777: Lane Equalization Preset Configuration 1 Register (n=0–1)
Offset: Port0: 0x000A06AC, Port1: 0x000A0EAC
Offset Formula: 0x000A06AC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x79B	Reserved

Table 1778: Lane Equalization Preset Configuration 2 Register (n=0–1)
Offset: Port0: 0x000A06B0, Port1: 0x000A0EB0
Offset Formula: 0x000A06B0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x7E2	Reserved

Table 1779: Lane Equalization Preset Configuration 3 Register (n=0–1)
Offset: Port0: 0x000A06B4, Port1: 0x000A0EB4
Offset Formula: 0x000A06B4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x61E	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1780: Lane Equalization Preset Configuration 4 Register (n=0–1)
Offset: Port0: 0x000A06B8, Port1: 0x000A0EB8
Offset Formula: 0x000A06B8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x71A	Reserved

Table 1781: Lane Equalization Preset Configuration 5 Register (n=0–1)
Offset: Port0: 0x000A06BC, Port1: 0x000A0EBC
Offset Formula: 0x000A06BC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x17	Reserved

Table 1782: Lane Equalization Preset Configuration 6 Register (n=0–1)
Offset: Port0: 0x000A06C0, Port1: 0x000A0EC0
Offset Formula: 0x000A06C0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x240	Reserved

Table 1783: Lane Equalization Preset Configuration 7 Register (n=0–1)
Offset: Port0: 0x000A06C4, Port1: 0x000A0EC4
Offset Formula: 0x000A06C4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x180	Reserved

Copyright © 2015 Marvell

Table 1784: Lane Equalization Preset Configuration 8 Register (n=0–1)
Offset: Port0: 0x000A06C8, Port1: 0x000A0EC8
Offset Formula: 0x000A06C8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x1C0	Reserved

Table 1785: Lane Equalization Preset Configuration 9 Register (n=0–1)
Offset: Port0: 0x000A06CC, Port1: 0x000A0ECC
Offset Formula: 0x000A06CC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x100	Reserved

Table 1786: Lane Equalization Preset Configuration 10 Register (n=0–1)
Offset: Port0: 0x000A06D0, Port1: 0x000A0ED0
Offset Formula: 0x000A06D0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x0	Reserved

Table 1787: Lane Equalization Preset Configuration 11 Register (n=0–1)
Offset: Port0: 0x000A06D4, Port1: 0x000A0ED4
Offset Formula: 0x000A06D4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x3	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1788: Lane Equalization Preset Configuration 12 Register (n=0–1)
Offset: Port0: 0x000A06D8, Port1: 0x000A0ED8
Offset Formula: 0x000A06D8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x4	Reserved

Table 1789: Lane Equalization Preset Configuration 13 Register (n=0–1)
Offset: Port0: 0x000A06DC, Port1: 0x000A0EDC
Offset Formula: 0x000A06DC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x1C3	Reserved

Table 1790: Lane Equalization Preset Configuration 14 Register (n=0–1)
Offset: Port0: 0x000A06E0, Port1: 0x000A0EE0
Offset Formula: 0x000A06E0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x104	Reserved

Table 1791: Lane Equalization Preset Configuration 15 Register (n=0–1)
Offset: Port0: 0x000A06E4, Port1: 0x000A0EE4
Offset Formula: 0x000A06E4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x6	Reserved

Copyright © 2015 Marvell

Table 1792: Lane Equalization Preset Configuration 16 Register (n=0–1)
Offset: Port0: 0x000A06E8, Port1: 0x000A0EE8
Offset Formula: 0x000A06E8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x2C0	Reserved

Table 1793: Lane Equalization Preset Configuration 17 Register (n=0–1)
Offset: Port0: 0x000A06EC, Port1: 0x000A0EEC
Offset Formula: 0x000A06EC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x0	Reserved

Table 1794: Lane Equalization Coefficient Max Setting 0 Register (n=0–1)
Offset: Port0: 0x000A06F0, Port1: 0x000A0EF0
Offset Formula: 0x000A06F0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x88E	Reserved

Table 1795: Lane Equalization Coefficient Max Setting 1 Register (n=0–1)
Offset: Port0: 0x000A06F4, Port1: 0x000A0EF4
Offset Formula: 0x000A06F4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x858E	Reserved

Copyright © 2015 Marvell

Table 1796: Lane Equalization Remote Setting Register (n=0–1)
Offset: Port0: 0x000A06F8, Port1: 0x000A0EF8
Offset Formula: 0x000A06F8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1797: Lane Equalization Status Register (n=0–1)
Offset: Port0: 0x000A06FC, Port1: 0x000A0EFC
Offset Formula: 0x000A06FC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1798: PIPE Revision Register (n=0–1)
Offset: Port0: 0x000A0700, Port1: 0x000A0F00
Offset Formula: 0x000A0700+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:8	MAIN_REVISION[7:0]	RO 0x20	Main-revision (PCIE PIPE PHY top revision ID)
7:0	SUB_REVISION[7:0]	RO 0x32	Sub-revision (PIPE revision ID)

Table 1799: Reset and Clock Control Register (n=0–1)
Offset: Port0: 0x000A0704, Port1: 0x000A0F04
Offset Formula: 0x000A0704+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9	MODE_CORE_CLK_FREQ_SEL	RW 0x0	CORE_CLK output frequency select, in mode_fixed_pclk=1, core_clk is always 250Mhz When mode_fixed_pclk = 0 0 = V250Mhz: 250Mhz 1 = V500Mhz: 500Mhz
8	PHY_RESET	RW 0x0	Common PHY reset 0 = normal operation 1 = common_phy reset
7	MODE_MULTICAST	RW 0x0	Select Multicast Register Mode 0 = normal mode 1 = write to a lane-specific register will result in a write to all lanes. The lane address bits will be ignored in this mode

Table 1799: Reset and Clock Control Register (n=0–1) (Continued)
Offset: Port0: 0x000A0704, Port1: 0x000A0F04
Offset Formula: 0x000A0704+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
6	MODE_CORE_CLK_CTRL	RW 0x0	CORE_CLK output control when power up or P2 state 0 = gate core_clk output 1 = un-gate core_clk output
5:4	MODE_REFDIV[1:0]	RW 0x2	Reference Clock Divisor This bit determines how sclk is generated from refclk_int 0 = divided by 1 1 = divided by 2 2 = divided by 4 3 = divided by 8
3	MODE_PIPE_WIDTH32	RW 0x1	PIPE Data bus width USB Mode 0 = 16-bit 0 = 16-bit @ 250MHz at 5.0GT/s 1 = 32-bit 1 = 32-bit @ 125MHz at 5.0GT/s
2	MODE_FIXED_PCLK	RW 0x0	PHY Datapath Width Mode -- PCIE 0 = 16-bit @125MHz at 2.5GT/s, 16-bit @ 250MHz at 5.0GT/s, 16bit @ 500Mhz at 8.0GT/s, 32-bit @62.5MHz at 2.5GT/s, 32-bit @ 125MHz at 5.0GT/s, 32bit @ 250Mhz at 8.0GT/s 1 = 8-bit @ 250MHz at 2.5GT/s, 16-bit @ 250MHz at 5.0GT/s, 320bit @250Mz at 8.0GT/s
1	REG_RESET	RW 0x0	PIPE Register Reset 0 = normal mode 1 = this bit resets all registers to default values (except the reset bits). Any register writes will have no effect
0	PIPE_SFT_RESET	RW 0x1	PIPE Soft Reset 0 = normal mode 1 = This bit is OR-ed with the power-on reset of the PHY. This bit does not affect register values

Table 1800: Test Mode Control Register (n=0–1)
Offset: Port0: 0x000A0708, Port1: 0x000A0F08
Offset Formula: 0x000A0708+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
14	MODE_STATE_OVERRIDE	RW 0x0	Override PM State Machine 0 = normal operation 1 = override PM state machine

Copyright © 2015 Marvell

Table 1800: Test Mode Control Register (n=0–1) (Continued)
Offset: Port0: 0x000A0708, Port1: 0x000A0F08
Offset Formula: 0x000A0708+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
13	MODE_RST_OVERRIDE	RW 0x0	Override Common PHY reset 0 = normal operation 1 = override Common PHY reset reset sequence is backward compatible with rev1.0
12	MODE_LB_SERDES	RW 0x0	Enable serdes loopback
11	MODE_LB_DEEP	RW 0x0	Enable deep loopback
10	MODE_LB_SHALLOW	RW 0x0	Enable shallow loopback
9	Reserved	RSVD 0x0	Reserved
8	DBG_TESTBUS_SEL[5]	RW 0x0	Test bus output enable 0 = test bus is disable at the lane 1 = test bus is enabled at the lane
7	DBG_TESTBUS_SEL[4]	RW 0x0	Select the per lane test bus to PHY_DBG_TEST_BUS 0 = select phy test bus 1 = select pipe test bus
6:3	DBG_TESTBUS_SEL[3:0]	RW 0x0	Select test bus lane 0 = pipe_testbus0 1 = pipe_testbus1 2 = pipe_testbus2 3 = pipe_testbus3 4 = pipe_testbus4 5 = pipe_testbus5 6 = pipe_testbus6 7 = pipe_testbus7 8 = pipe_testbus8 9 = pipe_testbus9 10 = pipe_testbusa 11 = pipe_testbusb 12 = pipe_testbusc 13 = VD: PHY testbus 14 = pipe_testbus 15 = pie_testbus
2	MODE_MARGIN_OVERRIDE	RW 0x0	Override Margining Controls from the MAC 0 = use margining signals from MAC 1 = use margining signals from lane configuration
1	MODE_PM_OVERRIDE	RW 0x0	Override PM Control Outputs 0 = normal operation 1 = override pm control outputs

Copyright © 2015 Marvell

Table 1800: Test Mode Control Register (n=0–1) (Continued)
Offset: Port0: 0x000A0708, Port1: 0x000A0F08
Offset Formula: 0x000A0708+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
0	MODE_BIST	RW 0x0	BIST Mode Enable When BIST mode is selected, common PCLK must be used. The BIST logic clock is enabled. 0 = normal mode 1 = BIST mode

Table 1801: Clock Source Low Register (n=0–1)
Offset: Port0: 0x000A070C, Port1: 0x000A0F0C
Offset Formula: 0x000A070C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:11	Reserved	RSVD 0x0	Reserved
10	MODE_P2_OFF	RW 0x0	p2 off mode enable. This bit control whether power management state macine goes into p2.off state when the mac drives the pipephy into p2 state and deassert mac_phy_clk_req_n signal, to exit p2.off. In p2.off state, reference clock can be removed; rxelectricalidle detector and tx common mode circuit are disabled. To exit p2.off state, PCIe system must asser perst and mac layer must assert the core_rst_n reset pin of the PIPE_PHY 0 = p2.off state is disabled 1 = p2.off state is enabled
9	CFG_USE_ALIGN_CLK	RW 0x0	Use lane_align_clock to reset txclk divider (source of pclk in32bit mode) 0 = do not use align clock 1 = use align clock
8	PRD_BUNDLE_PLL_RDY	RW 0x0	bundles pin_pll_ready_tx signals from all Comphys within the pcie link in multi-lane cases 1;b0: do not bundle pin_pll_ready_Tx signals 1;b0: bundle pin_pll_ready_Tx signals
7:5	PLL_READY_DLY[2:0]	RW 0x0	Delay ComPHY signal pin_pll_ready_tx before use it. Delay Time = (16 x pll_ready_dly x 2N) REFCLK cycles, where N = mode_refdiv, bit[5:4] of reg 0x1C1.
4	BUNDLE_SAMPLE_CTRL	RW 0x0	Bundle signal sampling control. 0 = sample bundle signals from local lane and external lanes at the same PCLK cycle 1 = sample bundle signals from local lane and external lanes at different PCLK cycles

Copyright © 2015 Marvell

Table 1801: Clock Source Low Register (n=0–1) (Continued)
Offset: Port0: 0x000A070C, Port1: 0x000A0F0C
Offset Formula: 0x000A070C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
3:2	BUNDLE_PERIOD_SCALE[1:0]	RW 0x0	Lane-to-lane bundle sampling period scale. When bundle_period_sel = 0 0 = V16ns: 16ns 0 = 4 PCLK cycles 1 = 8 PCLK cycles 1 = V32ns: 32ns 2 = 16 PCLK cycles 2 = V64ns: 64ns 3 = V128ns: 128ns 3 = 32 PCLK cycles
1	BUNDLE_PERIOD_SEL	RW 0x0	Select unit of lane-to-lane bundle clock sampling period. This is used for multi-lanes case only. This provides long enough time (multi-cycles) for lane-to-lane bundle logic (combinational logic) to meet setup time at layout. 0 = bundle sample period is per PCLK cycles 1 = bundle sample period is per fixed time (ns)
0	MODE_CLK_SRC	RW 0x1	This bit indicates the clock source of PCLK at this lane. 0 = PCLK is generated from REFCLK 1 = PCLK is generated from PLL

Table 1802: Clock Source High Register (n=0–1)
Offset: Port0: 0x000A0710, Port1: 0x000A0F10
Offset Formula: 0x000A0710+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	CFG_VBUS_PRESET_SRC	RW 0x0	USB3 Mode, VBUS present source This bit drives output port PHY_MAC_POWERPRESENT to indicate the presence of VBUS
14	CFG_RXTEM_ENABLE	RW 0x0	USB3 Mode, Receiver termination control select 0 = Termination control from MAC layer for all power states 1 = Termination control from MAC layer only at P3 state
13:9	Reserved	RSVD 0x0	Reserved
8	CFG_LANE_TURN_OFF_DIS	RW 0x0	Disable lane turned off signaling 0 = enable lane turned off signaling 1 = disable lane turned off signaling
7	MODE_PIPE4_IF	RW 0x0	PIPE version 4 mode enable 0 = disable PIPE version 4 mode 1 = enable PIPE version 4 mode

Copyright © 2015 Marvell

Table 1802: Clock Source High Register (n=0–1) (Continued)
Offset: Port0: 0x000A0710, Port1: 0x000A0F10
Offset Formula: 0x000A0710+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
6:5	Reserved	RSVD 0x0	Reserved
4:3	BIFURCATION_SEL [1:0]	RW 0x0	Birfurcation mux selection for internal version (1-bit) of SCLK_IN, LANE_TXCLK_REF_IN, and LANE_BUNDLE_CLK_REF_IN. 0 = select pins SCLK_IN[0], LANE_TXCLK_REF_IN[0], and LANE_BUNDLE_CLK_REF_IN[0]. 1 = select pins SCLK_IN[1], LANE_TXCLK_REF_IN[1], and LANE_BUNDLE_CLK_REF_IN[1]. 2 = select pins SCLK_IN[2], LANE_TXCLK_REF_IN[2], and LANE_BUNDLE_CLK_REF_IN[2]. 3 = reserved
2	LANE_MASTER	RW 0x1	The PCLK for the MAC must come from the master lane. 0 = indicates that tis lane is not the master lane of a link 1 = indicates that this lane is the master lane of a link
1	LANE_BREAK	RW 0x1	Indicates the highest order lane of a link 0 = indicates that this lane is the lower order lane of a link 1 = indicates that this lane is the highest order lane of a link
0	LANE_START	RW 0x1	Indicates the lowest order lane of a link 0 = indicates that this lane is the higher order lane of a link 1 = indicates that this lane is the lowest order lane of a link

Table 1803: Trigger Register (n=0–1)
Offset: Port0: 0x000A0714, Port1: 0x000A0F14
Offset Formula: 0x000A0714+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15	PULSE_DONE	RO 0x0	Trigger pulse status (for cfg_update, bist_start, counter_sample, or counter_sample_clear). 0 = trigger pulse is in progress 1 = trigger pulse process is done
14:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12	PMO_POWER_VALID	WO 0x0	When bit 1 of Reg. 0xC2 is 1, replace power state valid signal (the PHY uses it to sample pu_pll, pu_tx and pu_rx signals).
11	COUNTER_SAMPLE_CLEAR	WO 0x0	Sample and clear the counter (in sclk domain)
10	COUNTER_SAMPLE	WO 0x0	Sample counter, continue counting (in sclk domain)
9	BIST_START	WO 0x0	Trigger start of BIST sequence (in aux_clk domain)

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1803: Trigger Register (n=0–1) (Continued)
Offset: Port0: 0x000A0714, Port1: 0x000A0F14
Offset Formula: 0x000A0714+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
8	CFG_UPDATE	WO 0x0	Update PM registers (in sclk domain)
7:5	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
4:0	PULSE_LENGTH[4:0]	WO 0x0	Length of the trigger pulse in APB3 cycles (1-32)

Table 1804: Miscellaneous Control Register (n=0–1)
Offset: Port0: 0x000A0718, Port1: 0x000A0F18
Offset Formula: 0x000A0718+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15	CFG_CLK_ACK_TIMER_EN	RW 0x0	Clock request acknowledgement timer enable 0 = PIPE acknowledges the clock request to MAC when the PHY complete its refclk disable sequence. 1 = PIPE acknowledges the clock request to MAC when refclk_disable_dly[3:0] timer times out, or the PHY complete its refclk disable sequence, which ever happens first.
14	CFG_REFCLK_DET_TYPE	RW 0x0	REFCLK activity detection type. When this bit is set, user must configure target low frequency to be detected in GLOB_LOW_FREQ_CFG register. 0 = counter detect method 1 = low frequency detect method
13	MODE_REFCLK_DIS	RW 0x1	PHY REFCLK disable sequence enable. 0 = REFCLK disable sequence is not enabled 1 = REFCLK disable sequence is enabled
12	CFG_FREE_OSC_SEL	RW 0x0	Select free running osc clock as the source of PCLK during P1.CLKREQ state. 0 = PCLK is off 1 = PCLK is driven by OSCCLK
11	RCB_RXEN_SRC	RW 0x0	Receiver enable source of reference clock buffer

Copyright © 2015 Marvell

Table 1804: Miscellaneous Control Register (n=0–1) (Continued)
Offset: Port0: 0x000A0718, Port1: 0x000A0F18
Offset Formula: 0x000A0718+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
10:8	OSC_COUNT_SCALE[2:0]	RW 0x0	For Counter scale at OSCCLK domain for refclk_restore_dly timer and rxeidetect_dly timer. 0 = counts every 1 OSCCLK cycle 0 = counts every 1 OSCCLK cycle 1 = counts every 1 OSCCLK cycle 1 = counts every 2 OSCCLK cycles 2 = counts every 4 OSCCLK cycles 2 = counts every 1 OSCCLK cycle 3 = counts every 8 OSCCLK cycles 3 = counts every 1 OSCCLK cycle 4 = counts every 16 OSCCLK cycles 4 = counts every 4 OSCCLK cycles 5 = reserved 5 = reserved 6 = reserved 6 = reserved 7 = reserved 7 = reserved
7	MODE_P1_OFF	RW 0x1	PCIE MODE: P1 off mode enable This bit controls whether power management state machine goes into P1.OFF state when both PIPE signals mac_phy_rxeidetect_dis and mac_phy_txcmn_mode_dis are asserted during P1. 0 = dirve PHY_MAC_PWERPRESENT to 0 0 = P1.OFF state is disabled 1 = dirve PHY_MAC_PWERPRESENT to 1 1 = P1.OFF state is enabled
6	MODE_P1_SNOOZ	RW 0x1	P1 snooz mode enable This bit controls whether power management state machine goes into P1.SNOOZ state when PIPE signals mac_phy_rxeidetect_dis is asserted and mac_phy_txcmn_mode_dis remains low during P1. 0 = Termination control from MAC layer for all power states 0 = P1.SNOOZ state is disabled 1 = Termination control from MAC layer only at P3 state 1 = P1.SNOOZ state is enabled
5	CFG_RX_HIZ_SRC	RW 0x0	Control presence of receiver termination by register 0 = Rx Termination present 1 = Rx Termination removed
4	SQ_DETECT_OVERRIDE	RW 0x0	Override SQ_DETECTED input from Serdes 0 = normal operation (use SQ_DETECTED from Serdes 1 = use sq_detect_src as SQ_DETECTD signal source
3	SQ_DETECT_SRC	RW 0x0	SQ_DETECTED signal source

Copyright © 2015 Marvell

Table 1804: Miscellaneous Control Register (n=0–1) (Continued)
Offset: Port0: 0x000A0718, Port1: 0x000A0F18
Offset Formula: 0x000A0718+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
2	MODE_PCLK_CTRL	RW 0x0	PCLK output control. 0 = pclk is not inverted 1 = pclk is inverted
1	MODE_P2_PHYSTATUS	RW 0x0	PhyStatus behavior during P2 (PCIE mode) or P3 (USB3 mode) 0 = phy_mac_phystatus is low 1 = phy_mac_phystatus stays high
0	MODE_P1_CLK_REQ_N	RW 0x1	P1 clkreq mode enable (PCIE mode) This bit controls whether power management state machine goes into P1.CLKREQ state when mac_phy_clk_req_n is asserted during P1. 0 = P1.CLKREQ state is disabled 1 = P1.CLKREQ state is enabled

Table 1805: CLKREQ# Timing Control Register (n=0–1)
Offset: Port0: 0x000A071C, Port1: 0x000A0F1C
Offset Formula: 0x000A071C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	REFCLK_DISABLE_DLY[5:4]	RW 0x0	Receiver disable delay of reference clock buffer after Serdes has complete power down sequence and internal reference clock is stopped. This provides some refclk cycles to flush out the clock gating latency of Dphy and Serdes. Count at OSCCLK domain
13:10	REFCLK_DISABLE_DLY[3:0]	RW 0x5	Internal reference clock disable delay after MAC signal mac_phy_pclk_req_n is asserted. This ensures that the Serdes is completely powered down before relerence clock is stopped. Count at sclk domain with step size of 8 cycles
9:8	REFCLK_SHUTOFF_DLY[1:0]	RW 0x2	External REFCLK shut off (or output enable of CLKREQ# pad disabling) delay after disabling receiver of reference clock buffer. This provides some refclk cycles to flush out the disable latency. Count at OSCCLK domain
7:2	REFCLK_RESTORE_DLY[5:0]	RW 0xA	External REFCLK restore delay before re-enabling receiver of reference clock buffer. Count at OSCCLK domain.
1	CLKREQ_N_OVERRIDE	RW 0x0	Override CLKREQ# signal 0 = normal operation 1 = override CLKREQ# with register bit cfg_clkreq_n_src, and override refclk_rx_en with register bit rcb_rxen_src.
0	CLKREQ_N_SRC	RW 0x0	CLKREQ# signal source

Table 1806: Datapath Configuration Register (n=0–1)
Offset: Port0: 0x000A0720, Port1: 0x000A0F20
Offset Formula: 0x000A0720+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13	CFG_SAL_IGNORE_SQ	RW 0x0	Squelch detected signal handling on symbol alignment 0 = symbol alignment state machine is affected by squelch detected signal 1 = symbol alignment state machnie ignores squelch detected signal
12	CFG_TXELECIDLE_ASSERT	RW 0x0	ComPHY tx_idle_hiz assert Timing 0 = assert tx_idle_hiz two clock cycles later after sending the last symbol of EIOS 1 = assert tx_idle_hiz immediately after sending the last symbol of EIOS
11:10	CFG_GEN2_TXELECIDLE_DLY[1:0]	RW 0x0	ComPHY tx_idle_hiz Timing Gen2 0 = tx_idle_hiz is synchronous with txdata 1 = tx_idle_hiz is delayed by 1 clock cycle 2 = tx_idle_hiz is delayed by 2 clock cycle 3 = tx_idle_hiz is delayed by 3 clock cycle
9	CFG_SAL_FREEZE	RW 0x0	Symbol Alignment State Machine Freeze 0 = normal operation 1 = symbol alignment state machine stops. This mode is meant to be used temporarily while symbol alignment weights are being modified
8	CFG_ALWAYS_ALIGN	RW 0x0	Symbol Alignment Mode 0 = normal operation; symbol alignment is readjusted only when COM received and SYNC_FAIL state 1 = symbol alignment is always readjusted when COM symbol is received
7	CFG_DISABLE_SKP	RW 0x0	SKP Ordered Set Handling Mode 0 = normal operation 1 = SKP detection is disabled for the purposed of elastic buffer adjustment (overflow and underflow are still handled)
6	CFG_MASK_ERRORS	RW 0x0	8b/10b Decoder Error Masking 0 = normal operation 1 = 8b/10b decoder errors are masked off (only used for symbol alignment state machine)
5	CFG_DISABLE_EDB	RW 0x0	EDB Replacement of Error Symbols 0 = all symbols with 8b/10b errors are replaced by EDB 1 = symbols are never replaced by EDB
4	CFG_NO_DISPERROR	RW 0x0	Disparity Error Handling Mode 0 = disparity error is handled separately from 8b/10b error 1 = all decoder errors (including disparity errors) are handled as 8b/10b errors, including EDB replacement

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1806: Datapath Configuration Register (n=0–1) (Continued)
Offset: Port0: 0x000A0720, Port1: 0x000A0F20
Offset Formula: 0x000A0720+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
3	CFG_PASS_RXINFO	RW 0x0	Rx Data Mode 0 = rxdata, rxdatak and rxstatus are 0 when rxvalid=0 1 = rxdata, rxdatak and rxstatus keep values regardless of rxvalid
2:1	CFG_GEN1_TXELECIDLE_DLY[1:0]	RW 0x0	ComPHY tx_idle_hiz Timing Gen1 0 = tx_idle_hiz is synchronous with txdata 1 = tx_idle_hiz is delayed by 1 clock cycle 2 = tx_idle_hiz is delayed by 2 clock cycle 3 = tx_idle_hiz is delayed by 3 clock cycle
0	CFG_IGNORE_PHY_RDY	RW 0x0	ComPHY phy_rdy Handling 0 = positive edge on phy_rdy is required before rxvalid is asserted 1 = rxvalid logic ignored phy_rdy

Table 1807: Symbol Alignment Good Symbol Weight Register (n=0–1)
Offset: Port0: 0x000A0724, Port1: 0x000A0F24
Offset Formula: 0x000A0724+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x100	Reserved

Table 1808: Symbol Alignment Aligned COM Weight Register (n=0–1)
Offset: Port0: 0x000A0728, Port1: 0x000A0F28
Offset Formula: 0x000A0728+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x818	Reserved

Table 1809: Symbol Alignment Bad Symbol Weight Register (n=0–1)
Offset: Port0: 0x000A072C, Port1: 0x000A0F2C
Offset Formula: 0x000A072C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1809: Symbol Alignment Bad Symbol Weight Register (n=0–1) (Continued)
Offset: Port0: 0x000A072C, Port1: 0x000A0F2C
Offset Formula: 0x000A072C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x101F	Reserved

Table 1810: Symbol Alignment Misaligned COM Weight Register (n=0–1)
Offset: Port0: 0x000A0730, Port1: 0x000A0F30
Offset Formula: 0x000A0730+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:13	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
12:0	Reserved	RSVD 0x101F	Reserved

Table 1811: Symbol Alignment Squelch Detect Register (n=0–1)
Offset: Port0: 0x000A0734, Port1: 0x000A0F34
Offset Formula: 0x000A0734+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x202	Reserved

Table 1812: Symbol Alignment Comma Pattern Register (n=0–1)
Offset: Port0: 0x000A0738, Port1: 0x000A0F38
Offset Formula: 0x000A0738+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:10	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
9:0	Reserved	RSVD 0xFA	Reserved

Copyright © 2015 Marvell

Table 1813: Power Management Timing Parameter 1 Register (n=0–1)
Offset: Port0: 0x000A0740, Port1: 0x000A0F40
Offset Formula: 0x000A0740+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	CFG_PM_OSCCLK_WAIT[3:0]	RW 0x0	Amount of time spent in PLLON power state waiting for the completion of the last oscclk pulse on pclk/aux_clk
11:8	CFG_PM_RXDEN_WAIT[3:0]	RW 0x1	Amount of time spent in the decision state (to determine if rx present or not) after seeing internal signal TXDETRX_VALID asserted. PCIE mode: When bit 6 of register 0x8B is 0, in units of 0.32us; When bit 6 of register 0x8B is 1, in units of 40.96us USB3 mode: When bit 6 of register 0x8B is 0, not apply; When bit 6 of register 0x8B is 1, in units of 32.768us
7:0	CFG_PM_RXDLOZ_WAIT[7:0]	RW 0x1E	Amount of time spent in the LoZ state before receiver detection is initiated. PCIE mode: In units of 5.12us (for sclk = REFCLK/4, where REFCLK = 100MHz) USB3 mode: In units of 2.048us

Table 1814: Power Management Timing Parameter 2 Register (n=0–1)
Offset: Port0: 0x000A0744, Port1: 0x000A0F44
Offset Formula: 0x000A0744+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x3014	Reserved

Table 1815: Counter Lane and Type Register (n=0–1)
Offset: Port0: 0x000A0748, Port1: 0x000A0F48
Offset Formula: 0x000A0748+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13	PMO_REFCLK_DIS	RW 0x0	Replace dphy_ana_refclk_dis when reg 0x1c2[1]=1, control disabling refclk at comphy 0 = normal operation 1 = request to disable refclk at comphy

Copyright © 2015 Marvell

Table 1815: Counter Lane and Type Register (n=0–1) (Continued)
Offset: Port0: 0x000A0748, Port1: 0x000A0F48
Offset Formula: 0x000A0748+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
12	PMO_PU_SQ	RW 0x0	Replace dphy_ana_pu_sq when reg 0x1c2[1] =1. power up comphy squelch detector 0 = power down 1 = power up
11:8	COUNTER_TYPE[3:0]	RW 0x0	Counter Type 0 = 8b/10b error 1 = disparity error 2 = 8b/10b or disparity error 3 = 8b/10b error (only during rxvalid) 4 = disparity error (only during rxvalid) 5 = 8b/10b or disparity error (only during rxvalid) 6 = rxvalid=1 7 = COM-SKP pairs 8 = SKP added 9 = SKP deleted 10 = underflow 11 = overflow 12 = VC: rxvalid transitions 13 = VD: reserved 14 = VE: reserved 15 = VF: reserved
7:5	Reserved	RSVD 0x0	Reserved
4:0	COUNTER_LANE[4:0]	RW 0x0	set to 5'h0

Table 1816: Counter Low Register (n=0–1)
Offset: Port0: 0x000A074C, Port1: 0x000A0F4C
Offset Formula: 0x000A074C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:0	COUNTER_SAMPLED[15:0]	RO 0x0	low 16 bits of the sampled counter value

Table 1817: Counter High Register (n=0–1)
Offset: Port0: 0x000A0750, Port1: 0x000A0F50
Offset Formula: 0x000A0750+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	COUNTER_SAMPLED[31:16]	RO 0x0	high 16 bits of the sampled counter value

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 1818: PM Serdes Control Override Register (n=0–1)
Offset: Port0: 0x000A0754, Port1: 0x000A0F54
Offset Formula: 0x000A0754+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x1	Reserved

Table 1819: PM Datapath and Clock Control Override Register (n=0–1)
Offset: Port0: 0x000A0758, Port1: 0x000A0F58
Offset Formula: 0x000A0758+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1820: Low Frequency Configuration Register (n=0–1)
Offset: Port0: 0x000A075C, Port1: 0x000A0F5C
Offset Formula: 0x000A075C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	LOW_FREQ_CNT_SCALE[1:0]	RW 0x0	Counter scale at REFCLK domain for target low frequency clock period counting during REFCLK activity detection. 0 = counts every 1 REFCLK cycle 1 = counts every 10 REFCLK cycles 2 = counts every 50 REFCLK cycles 3 = counts every 100 REFCLK cycles
13:7	LOW_FREQ_PERIOD_MAX[6:0]	RW 0x32	Maximum target low frequency clock (OSCCLK) period cycles counted by REFCLK.
6:0	LOW_FREQ_PERIOD_MIN[6:0]	RW 0x30	Minimum target low frequency clock (OSCCLK) period cycles counted by REFCLK.

Table 1821: Block Alignment Good Block Weight Register (n=0–1)
Offset: Port0: 0x000A0760, Port1: 0x000A0F60
Offset Formula: 0x000A0760+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x100	Reserved

Copyright © 2015 Marvell

Table 1822: Block Alignment Aligned EIEOS Weight Register (n=0–1)
Offset: Port0: 0x000A0764, Port1: 0x000A0F64
Offset Formula: 0x000A0764+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x818	Reserved

Table 1823: Block Alignment Bad Block Weight Register (n=0–1)
Offset: Port0: 0x000A0768, Port1: 0x000A0F68
Offset Formula: 0x000A0768+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x101F	Reserved

Table 1824: Block Alignment Misaligned EIEOS Weight Register (n=0–1)
Offset: Port0: 0x000A076C, Port1: 0x000A0F6C
Offset Formula: 0x000A076C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:14	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
13:0	Reserved	RSVD 0x101F	Reserved

Table 1825: Block Alignment Squelch Detect Register (n=0–1)
Offset: Port0: 0x000A0770, Port1: 0x000A0F70
Offset Formula: 0x000A0770+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:12	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
11:0	Reserved	RSVD 0x202	Reserved

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1826: BIST Control Input Register (n=0–1)
Offset: Port0: 0x000A0780, Port1: 0x000A0F80
Offset Formula: 0x000A0780+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:11	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
10	BIST_CLK_REQ_N	RW 0x0	CLKREQ control when Reg 0x1C2[0] = 1
9:8	BIST_TXDATAK[1:0]	RW 0x0	also modified by the BIST generator
7	BIST_RXPOLARITY	RW 0x0	Select RX polarity inversion when Reg 0x1C2[0] = 1 0 = no polarity inversion 1 = polaity inversion
6	BIST_TXCOMPLIANCE	RW 0x0	Control TX compliance signaling when Reg 0x1C2[0] = 1
5	BIST_TXELECIDLE	RW 0x1	Control TX electrical idle when Reg 0x1C2[0] = 1 0 = sending data or beacon 1 = set transmitter to electrical idle
4	BIST_TXDETECTRX_LOOPBACK	RW 0x0	Control TX receiver detection or loopback when Reg 0x1C2[0] = 1 0 = normal operation 1 = trigger tx receiver detection or rx to tx loopback
3:2	BIST_RATE[1:0]	RW 0x0	Control signal rate when Reg 0x1C2[0] = 1 0 = use 2.5GT/s signal rate 1 = use 5.0GT/s signal rate 2 = use 8.0GT/s signal rate 3 = reserved
1:0	BIST_POWERDOWN[1:0]	RW 0x2	Control power state when Reg 0x1C2[0] = 1 0 = P0, normal operation 1 = P0s, power saving state, low recovery time latency 2 = P1, lower power state, longer recovery time latency 3 = P2, lowest power state, longest recovery time latency

Table 1827: BIST Configuration Input Register (n=0–1)
Offset: Port0: 0x000A0784, Port1: 0x000A0F84
Offset Formula: 0x000A0784+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:9	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.

Table 1827: BIST Configuration Input Register (n=0–1) (Continued)
Offset: Port0: 0x000A0784, Port1: 0x000A0F84
Offset Formula: 0x000A0784+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
8:0	Reserved	RSVD 0x0	Reserved

Table 1828: BIST Data Input Register (n=0–1)
Offset: Port0: 0x000A0788, Port1: 0x000A0F88
Offset Formula: 0x000A0788+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1829: BIST LANE and TYPE Register (n=0–1)
Offset: Port0: 0x000A078C, Port1: 0x000A0F8C
Offset Formula: 0x000A078C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:14	Reserved	RSVD 0x0	Reserved
13:3	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
2:0	Reserved	RSVD 0x0	Reserved

Table 1830: BIST Loggin Window Start Cycle Register (n=0–1)
Offset: Port0: 0x000A0790, Port1: 0x000A0F90
Offset Formula: 0x000A0790+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x0	Reserved

Table 1831: BIST Loggin Window Length Register (n=0–1)
Offset: Port0: 0x000A0794, Port1: 0x000A0F94
Offset Formula: 0x000A0794+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x1	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1832: BIST Result Mask Low Register (n=0–1)
Offset: Port0: 0x000A0798, Port1: 0x000A0F98
Offset Formula: 0x000A0798+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0xFFFF	Reserved

Table 1833: BIST Result Mask High Register (n=0–1)
Offset: Port0: 0x000A079C, Port1: 0x000A0F9C
Offset Formula: 0x000A079C+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0xFFFF	Reserved

Table 1834: BIST Result Low Register (n=0–1)
Offset: Port0: 0x000A07A0, Port1: 0x000A0FA0
Offset Formula: 0x000A07A0+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1835: BIST Result High Register (n=0–1)
Offset: Port0: 0x000A07A4, Port1: 0x000A0FA4
Offset Formula: 0x000A07A4+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Table 1836: BIST Sequencer Configuration Register (n=0–1)
Offset: Port0: 0x000A07A8, Port1: 0x000A0FA8
Offset Formula: 0x000A07A8+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1837: BIST LFSR Seed Register (n=0–1)
Offset: Port0: 0x000A07AC, Port1: 0x000A0FAC
Offset Formula: 0x000A07AC+0x800*n: where n (0-1) represents Port

Bit	Field	Type/InitVal	Description
15:8	reserved	RO 0x0	Reserved Reserved. Always write 0. Ignore read value.
7:0	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

A.25 XOR Engine Registers

The following table provides a summarized list of all registers that belong to the XOR Engine, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1838: Summary Map Table for the XOR Engine Registers

Register Name	Offset	Table and Page
XOR Engine Address Decoding		
XOR Engine Window Control (XEnWCR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060A40, Xor Engine0XOR Channel1: 0x00060A44, Xor Engine1XOR Channel0: 0x00060B40, Xor Engine1XOR Channel1: 0x00060B44	Table 1839, p. 1366
XOR Engine Base Address (XEBARx)<n> Register (m=0–1, n=0–7)	Xor Engine0XEBAR0: 0x00060A50, Xor Engine0XEBAR1: 0x00060A54...Xor Engine1XEBAR7: 0x00060B6C	Table 1840, p. 1367
XOR Engine Size Mask (XESMRx)<n> Register (m=0–1, n=0–7)	Xor Engine0XESMR0: 0x00060A70, Xor Engine0XESMR1: 0x00060A74...Xor Engine1XESMR7: 0x00060B8C	Table 1841, p. 1367
XOR Engine High Address Remap (XEHARRx)<n> Register (m=0–1, n=0–3)	Xor Engine0XEHARR0: 0x00060A90, Xor Engine0XEHARR1: 0x00060A94, Xor Engine0XEHARR2: 0x00060A98, Xor Engine0XEHARR3: 0x00060A9C, Xor Engine1XEHARR0: 0x00060B90, Xor Engine1XEHARR1: 0x00060B94, Xor Engine1XEHARR2: 0x00060B98, Xor Engine1XEHARR3: 0x00060B9C	Table 1842, p. 1368
XOR Engine Address Override Control (XEAOCR)<n> Register (m=0–1, n=0–1)	Xor Engine0XEAOCR0: 0x00060AA0, Xor Engine0XEAOCR1: 0x00060AA4, Xor Engine1XEAOCR0: 0x00060BA0, Xor Engine1XEAOCR1: 0x00060BA4	Table 1843, p. 1368
XOR Engine Control		
XOR Engine Channel Arbiter (XECHAR) Register (m=0–1)	Xor Engine0: 0x00060800, Xor Engine1: 0x00060900	Table 1844, p. 1370

Table 1838: Summary Map Table for the XOR Engine Registers (Continued)

Register Name	Offset	Table and Page
XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060810, Xor Engine0XOR Channel1: 0x00060814, Xor Engine1XOR Channel0: 0x00060910, Xor Engine1XOR Channel1: 0x00060914	Table 1845, p. 1371
XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060820, Xor Engine0XOR Channel1: 0x00060824, Xor Engine1XOR Channel0: 0x00060920, Xor Engine1XOR Channel1: 0x00060924	Table 1846, p. 1372
XOR Engine Outstanding Reads (XEnORDS)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060880, Xor Engine0XOR Channel1: 0x00060884, Xor Engine1XOR Channel0: 0x00060980, Xor Engine1XOR Channel1: 0x00060984	Table 1847, p. 1373
XOR Engine Descriptor Registers		
XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060A00, Xor Engine0XOR Channel1: 0x00060A04, Xor Engine1XOR Channel0: 0x00060B00, Xor Engine1XOR Channel1: 0x00060B04	Table 1848, p. 1374
XOR Engine Current Descriptor Pointer (XEnCDPR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060A10, Xor Engine0XOR Channel1: 0x00060A14, Xor Engine1XOR Channel0: 0x00060B10, Xor Engine1XOR Channel1: 0x00060B14	Table 1849, p. 1374
XOR Engine Byte Count (XEnBCR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060A20, Xor Engine0XOR Channel1: 0x00060A24, Xor Engine1XOR Channel0: 0x00060B20, Xor Engine1XOR Channel1: 0x00060B24	Table 1850, p. 1374
XOR Engine Memory Initialization		

Copyright © 2015 Marvell

Table 1838: Summary Map Table for the XOR Engine Registers (Continued)

Register Name	Offset	Table and Page
XOR Engine Destination Pointer (XEnDPR0)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060AB0, Xor Engine0XOR Channel1: 0x00060AB4, Xor Engine1XOR Channel0: 0x00060BB0, Xor Engine1XOR Channel1: 0x00060BB4	Table 1851, p. 1375
XOR Engine Block Size (XEnBSR)<n> Register (m=0–1, n=0–1)	Xor Engine0XOR Channel0: 0x00060AC0, Xor Engine0XOR Channel1: 0x00060AC4, Xor Engine1XOR Channel0: 0x00060BC0, Xor Engine1XOR Channel1: 0x00060BC4	Table 1852, p. 1375
XOR Engine Timer Mode Control (XETMCR) Register (m=0–1)	Xor Engine0: 0x00060AD0, Xor Engine1: 0x00060BD0	Table 1853, p. 1375
XOR Engine Timer Mode Initial Value (XETMIVR) Register (m=0–1)	Xor Engine0: 0x00060AD4, Xor Engine1: 0x00060BD4	Table 1854, p. 1376
XOR Engine Timer Mode Current Value (XETMCVR) Register (m=0–1)	Xor Engine0: 0x00060AD8, Xor Engine1: 0x00060BD8	Table 1855, p. 1376
XOR Engine Initial Value Low (XEIVRL) Register (m=0–1)	Xor Engine0: 0x00060AE0, Xor Engine1: 0x00060BE0	Table 1856, p. 1377
XOR Engine Initial Value High (XEIVRH) Register (m=0–1)	Xor Engine0: 0x00060AE4, Xor Engine1: 0x00060BE4	Table 1857, p. 1377
XOR Engine Interrupt Registers		
XOR Engine Interrupt Cause (XEICR1) Register (m=0–1)	Xor Engine0: 0x00060830, Xor Engine1: 0x00060930	Table 1858, p. 1377
XOR Engine Interrupt Mask (XEIMR) Register (m=0–1)	Xor Engine0: 0x00060840, Xor Engine1: 0x00060940	Table 1859, p. 1380
XOR Engine Error Cause (XEECR) Register (m=0–1)	Xor Engine0: 0x00060850, Xor Engine1: 0x00060950	Table 1860, p. 1380
XOR Engine Error Address (XEEAR) Register (m=0–1)	Xor Engine0: 0x00060860, Xor Engine1: 0x00060960	Table 1861, p. 1381

Copyright © 2015 Marvell

A.25.1 XOR Engine Address Decoding

Table 1839: XOR Engine Window Control (XEnWCR)<n> Register (m=0–1, n=0–1)
Offset: Xor Engine0XOR Channel0: 0x00060A40, Xor Engine0XOR Channel1: 0x00060A44, Xor Engine1XOR Channel0: 0x00060B40, Xor Engine1XOR Channel1: 0x00060B44
Offset Formula: 0x00060A40+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:30	Win7acc	RW 0x3	Window7 Access Control See Window0 Access Control.
29:28	Win6acc	RW 0x3	Window6 Access Control See Window0 Access Control.
27:26	Win5acc	RW 0x3	Window5 Access Control See Window0 Access Control.
25:24	Win4acc	RW 0x3	Window4 Access Control See Window0 Access Control.
23:22	Win3acc	RW 0x3	Window3 Access Control See Window0 Access Control.
21:20	Win2acc	RW 0x3	Window2 Access Control See Window0 Access Control.
19:18	Win1acc	RW 0x3	Window1 Access Control See Window0 Access Control.
17:16	Win0acc	RW 0x3	Window0 Access Control In case of a write protect violation (for example, writing of data to a read only region), an interrupt is set, and the transaction is not driven to the target interface. 0 = No access allowed 1 = Read only 2 = Reserved 3 = Full access: Full Read or Write access
15:8	Reserved	RO 0x0	Reserved
7	Win7en	RW 0x0	Window7 Enable See Window0 Enable.
6	Win6en	RW 0x0	Window6 Enable See Window0 Enable.
5	Win5en	RW 0x0	Window5 Enable See Window0 Enable.
4	Win4en	RW 0x0	Window4 Enable See Window0 Enable.
3	Win3en	RW 0x0	Window3 Enable See Window0 Enable.
2	Win2en	RW 0x0	Window2 Enable See Window0 Enable.
1	Win1en	RW 0x0	Window1 Enable See Window0 Enable.

Copyright © 2015 Marvell

Table 1839: XOR Engine Window Control (XEnWCR)<n> Register (m=0–1, n=0–1) (Continued)

Offset: Xor Engine0XOR Channel0: 0x00060A40, Xor Engine0XOR Channel1: 0x00060A44, Xor Engine1XOR Channel0: 0x00060B40, Xor Engine1XOR Channel1: 0x00060B44

Offset Formula: 0x00060A40+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
0	Win0en	RW 0x0	Window0 Enable 0 = Disable 1 = Enable

Table 1840: XOR Engine Base Address (XEBARx)<n> Register (m=0–1, n=0–7)

Offset: Xor Engine0XEBAR0: 0x00060A50, Xor Engine0XEBAR1: 0x00060A54...Xor Engine1XEBAR7: 0x00060B6C

Offset Formula: 0x00060A50+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-7) represents XEBAR

Bit	Field	Type/InitVal	Description
31:16	Base	RW 0x0	Base Address Used with the corresponding XOR Engine Size Mask register to set the address window size and location within the range of 4 GB space. The base address granularity is 64 KB.
15:8	Attr	RW 0x0	Specifies target specific attributes depending on the target interface.
7:4	Reserved	RO 0x0	Reserved
3:0	Target	RW 0x0	Specifies the target interface associated with this window. See Address Decoding section for full details.

Table 1841: XOR Engine Size Mask (XESMRx)<n> Register (m=0–1, n=0–7)

Offset: Xor Engine0XESMR0: 0x00060A70, Xor Engine0XESMR1: 0x00060A74...Xor Engine1XESMR7: 0x00060B8C

Offset Formula: 0x00060A70+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-7) represents XESMR

Bit	Field	Type/InitVal	Description
31:16	SizeMask	RW 0x0	Window Size Used with the corresponding XOR Engine Base Address register to set the address window size and location within the range of the 4 GB space. Must be programmed from LSB to MSB as sequence of 1s followed by sequence of 0s. The number of 1s specifies the size of the window in 64 KB granularity (for example, a value of 0x00ff specifies 256x64k = 16 MB).
15:0	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 1842: XOR Engine High Address Remap (XEHARRx)<n> Register (m=0–1, n=0–3)

Offset: Xor Engine0XEHARR0: 0x00060A90, Xor Engine0XEHARR1: 0x00060A94, Xor Engine0XEHARR2: 0x00060A98, Xor Engine0XEHARR3: 0x00060A9C, Xor Engine1XEHARR0: 0x00060B90, Xor Engine1XEHARR1: 0x00060B94, Xor Engine1XEHARR2: 0x00060B98, Xor Engine1XEHARR3: 0x00060B9C

Offset Formula: 0x00060A90+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-3) represents XEHARR

Bit	Field	Type/InitVal	Description
NOTE: High Address Remap Register #N corresponds to Base Address register #N, respectively.			
31:0	Remap	RW 0x0	Remap Address Specifies address bits[63:32] to be driven to the target interface for the corresponding window (window0 to window3). Only relevant for target interfaces that supports more than a 4 GB address space. When using target interface that do not support more than a 4 GB address space, this register must be cleared. **NOTE:** For window4 to window7, 0x0 will be generated to address bits [63:32].

Table 1843: XOR Engine Address Override Control (XEAOCR)<n> Register (m=0–1, n=0–1)

Offset: Xor Engine0XEAOCR0: 0x00060AA0, Xor Engine0XEAOCR1: 0x00060AA4, Xor Engine1XEAOCR0: 0x00060BA0, Xor Engine1XEAOCR1: 0x00060BA4

Offset Formula: 0x00060AA0+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XEAOCR

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:28	NDAOvrPtr	RW 0x0	Override Next Descriptor Address Pointer Specifies the register from which the override parameters will be taken. **NOTE:** Valid only if NDAOvrEn is set. 0 = XEHARR0: Target and attributes are taken from XEBAR0. Address[63:32] taken from XEHARR0. 1 = XEHARR1: Target and attributes are taken from XEBAR1. Address[63:32] taken from XEHARR1. 2 = XEHARR2: Target and attributes are taken from XEBAR2. Address[63:32] taken from XEHARR2. 3 = XEHARR3: Target and attributes are taken from XEBAR3. Address[63:32] taken from XEHARR3.
27	NDAOvrEn	RW 0x0	Override Next Descriptor Address Control 0 = No Override 1 = Override enable

Copyright © 2015 Marvell

Table 1843: XOR Engine Address Override Control (XEAOCR)<n> Register (m=0–1, n=0–1) (Continued)
Offset: Xor Engine0XEAOCR0: 0x00060AA0, Xor Engine0XEAOCR1: 0x00060AA4, Xor Engine1XEAOCR0: 0x00060BA0, Xor Engine1XEAOCR1: 0x00060BA4
Offset Formula: 0x00060AA0+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XEAOCR

Bit	Field	Type/InitVal	Description
26:25	DAOvrPtr	RW 0x0	Override Destination/P and Q Address Pointer Specifies the register from which the override parameters will be taken. **NOTE:** Valid only if DAOvrEn is set. 0 = XEHARR0: Target and attributes are taken from XEBAR0. Address[63:32] taken from XEHARR0. 1 = XEHARR1: Target and attributes are taken from XEBAR1. Address[63:32] taken from XEHARR1. 2 = XEHARR2: Target and attributes are taken from XEBAR2. Address[63:32] taken from XEHARR2. 3 = XEHARR3: Target and attributes are taken from XEBAR3. Address[63:32] taken from XEHARR3.
24	DAOvrEn	RW 0x0	Override Destination/P and Q Address Control 0 = No override 1 = Override enable
23:22	SA7OvrPtr	RW 0x0	Override Source Address #7 Pointer **NOTE:** Valid only if SA7OvrEn is set.
21	SA7OvrEn	RW 0x0	Override Source Address #7 Control
20:19	SA6OvrPtr	RW 0x0	Override Source Address #6 Pointer **NOTE:** Valid only if SA6OvrEn is set.
18	SA6OvrEn	RW 0x0	Override Source Address #6 Control
17:16	SA5OvrPtr	RW 0x0	Override Source Address #5 Pointer **NOTE:** Valid only if SA5OvrEn is set.
15	SA5OvrEn	RW 0x0	Override Source Address #5 Control
14:13	SA4OvrPtr	RW 0x0	Override Source Address #4 Pointer **NOTE:** Valid only if SA4OvrEn is set.
12	SA4OvrEn	RW 0x0	Override Source Address #4 Control
11:10	SA3OvrPtr	RW 0x0	Override Source Address #3 Pointer **NOTE:** Valid only if SA3OvrEn is set.
9	SA3OvrEn	RW 0x0	Override Source Address #3 Control
8:7	SA2OvrPtr	RW 0x0	Override Source Address #2 Pointer **NOTE:** Valid only if SA2OvrEn is set.
6	SA2OvrEn	RW 0x0	Override Source Address #2 Control
5:4	SA1OvrPtr	RW 0x0	Override Source Address #1 Pointer **NOTE:** Valid only if SA1OvrEn is set.
3	SA1OvrEn	RW 0x0	Override Source Address #1 Control

Copyright © 2015 Marvell

Table 1843: XOR Engine Address Override Control (XEAOCR)<n> Register (m=0–1, n=0–1) (Continued)
Offset: Xor Engine0XEAOCR0: 0x00060AA0, Xor Engine0XEAOCR1: 0x00060AA4, Xor Engine1XEAOCR0: 0x00060BA0, Xor Engine1XEAOCR1: 0x00060BA4
Offset Formula: 0x00060AA0+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XEAOCR

Bit	Field	Type/InitVal	Description
2:1	SA0OvrPtr	RW 0x0	Override Source Address #0 Pointer Specifies the register from which the override parameters will be taken. **NOTE:** Valid only if SA0OvrEn is set. 0 = XEHARR0: Target and attributes are taken from XEBAR0. Address[63:32] taken from XEHARR0. 1 = XEHARR1: Target and attributes are taken from XEBAR1. Address[63:32] taken from XEHARR1. 2 = XEHARR2: Target and attributes are taken from XEBAR2. Address[63:32] taken from XEHARR2. 3 = XEHARR3: Target and attributes are taken from XEBAR3. Address[63:32] taken from XEHARR3.
0	SA0OvrEn	RW 0x0	Override Source Address #0 Control

A.25.2 XOR Engine Control

Table 1844: XOR Engine Channel Arbiter (XECHAR) Register (m=0–1)
Offset: Xor Engine0: 0x00060800, Xor Engine1: 0x00060900
Offset Formula: 0x00060800+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:8	Reserved	RO 0x0	Reserved
7	Slice7	RW 0x1	Slice #7 of the channel pizza arbiter
6	Slice6	RW 0x0	Slice #6 of the channel pizza arbiter
5	Slice5	RW 0x1	Slice #5 of the channel pizza arbiter
4	Slice4	RW 0x0	Slice #4 of the channel pizza arbiter
3	Slice3	RW 0x1	Slice #3 of the channel pizza arbiter
2	Slice2	RW 0x0	Slice #2 of the channel pizza arbiter
1	Slice1	RW 0x1	Slice #1 of the channel pizza arbiter
0	Slice0	RW 0x0	Slice #0 of the channel pizza arbiter. 0 = Channel0: Slice is owned by channel 0. 1 = Channel1: Slice is owned by channel 1.

Copyright © 2015 Marvell
Document Classification: Proprietary Information
November 25, 2015, Preliminary

Table 1845: XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1)

Offset: Xor Engine0XOR Channel0: 0x00060810, Xor Engine0XOR Channel1: 0x00060814, Xor Engine1XOR Channel0: 0x00060910, Xor Engine1XOR Channel1: 0x00060914

Offset Formula: 0x00060810+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:16	Reserved	RO 0x0	Reserved
15	RegAccProtect	RW 0x1	Internal Register Access Protection Enable 0 = Disable 1 = Enable
14	DesSwp	RW 0x0	Descriptor Read/Write Endianness Swap Control If swapping is enabled, byte 0 is exchanged with byte 7, byte 1 with byte 6 etc. 0 = No swap 1 = Swap
13	DwrReqSwp	RW 0x0	Data Write Request Endianness Swap Control If swapping is enabled, byte 0 is exchanged with byte 7, byte 1 with byte 6 etc. 0 = No swap 1 = Swap
12	DrdResSwp	RW 0x0	Data Read Response Endianness Swap Control If swapping is enabled, byte 0 is exchanged with byte 7, byte 1 with byte 6 etc. 0 = No Swap 1 = Swap
11	Reserved	RW 0x0	Reserved
10:8	DstBurstLimit	RW 0x4	Burst Limit in each destination write request access over the internal Mbus Note: When the RAID6_recovery mode of operation, as defined in the descriptor, is used, then for P & Q buffers DstBurstLimit field is used and for all the source buffers SrcBurstLimit field is used. 2 = 32 Bytes 3 = 64 Bytes 4 = 128 Bytes
7	Reserved	RW 0x0	Reserved
6:4	SrcBurstLimit	RW 0x4	Burst Limit in each source read request access over the internal Mbus Note: When the RAID6_recovery mode of operation, as defined in the descriptor, is used, then for P & Q buffers DstBurstLimit field is used and for all the source buffers SrcBurstLimit field is used. 2 = 32 Bytes 3 = 64 Bytes 4 = 128 Bytes
3	Reserved	RW 0x0	Reserved

Copyright © 2015 Marvell

Table 1845: XOR Engine Configuration (XEnCR)<n> Register (m=0–1, n=0–1) (Continued)

Offset: Xor Engine0XOR Channel0: 0x00060810, Xor Engine0XOR Channel1: 0x00060814, Xor Engine1XOR Channel0: 0x00060910, Xor Engine1XOR Channel1: 0x00060914

Offset Formula: 0x00060810+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
2:0	OperationMode	RW 0x0	Specifies the type of operation to be carried out by XOR Engine. Note: When using the "Descriptor" mode of operation, then all descriptors should be aligned to 64 byte Note: When using the "Descriptor" mode of operation, then the descriptors of a CRC-32 chain should not be interleaved with descriptors of other mode of operation 0 = XOR calculate operation 1 = CRC-32 calculate operation 2 = DMA operation 3 = ECC_cleanup_operation 4 = Memory Initialization operation 7 = Descriptor: The operation mode is as defined in the descriptor

Table 1846: XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1)

Offset: Xor Engine0XOR Channel0: 0x00060820, Xor Engine0XOR Channel1: 0x00060824, Xor Engine1XOR Channel0: 0x00060920, Xor Engine1XOR Channel1: 0x00060924

Offset Formula: 0x00060820+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:6	Reserved	RO 0x0	Reserved
5:4	XEstatus	RO 0x0	XOR Engine Status Indication 0 = Channel not active 1 = Channel active 2 = Channel paused 3 = Reserved
3	XErestart	WO 0x0	XOR Engine Restart after Pause Control 0 = No meaning: Clearing this bit has no meaning and will be disregarded by the XOR Engine. 1 = Pause state: The XOR Engine restart after pause. Setting this bit after the XOR Engine channel enters the pause state re-activates the channel and resumes the suspended operation execution.
2	XEpause	WO 0x0	XOR Engine Pause Control 0 = No meaning: Clearing this bit has no meaning and will be disregarded by XOR Engine. 1 = Pause state: When the software sets this bit, it pauses the relevant XOR Engine channel. XOR Engine will suspend at the earliest opportunity (refer to Pause Operation section of the Functional Specification). After entering paused state, the XOR Engine will signal the software by setting the <XEstatus> field to Channel paused (0x2) and asserting the paused interrupt.

Copyright © 2015 Marvell

Table 1846: XOR Engine Activation (XEnACTR)<n> Register (m=0–1, n=0–1) (Continued)

Offset: Xor Engine0XOR Channel0: 0x00060820, Xor Engine0XOR Channel1: 0x00060824, Xor Engine1XOR Channel0: 0x00060920, Xor Engine1XOR Channel1: 0x00060924

Offset Formula: 0x00060820+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
1	XEstop	WO 0x0	**NOTE:** Setting the <XEstop> field when the XOR Engine is inactive or paused will be disregarded. 0 = No meaning: Clearing this bit has no meaning and will be disregarded by XOR Engine. 1 = Stop: When the software sets this bit, it de-activates the relevant XOR Engine channel. XOR Engine will stop the current operation at the earliest opportunity (refer to Stop Operation section of the Functional Specification). After entering de-active state the XOR Engine will signal the software by setting the <XEstatus> field to Channel not active (0x0) and asserting the stopped interrupt.
0	XEStart	WO 0x0	**NOTE:** Software must confirm that the <XEstatus> field is in the Channel not active (0x0) state. Setting <XEstart> when XOR Engine is active will be disregarded. 0 = No meaning: Clearing this bit has no meaning and will be disregarded by XOR Engine. 1 = Start: When the software sets this bit, it activates the relevant XOR Engine channel. After entering active state, XOR Engine will signal the software by setting the <XEstatus> field to Channel active (0x1).

Table 1847: XOR Engine Outstanding Reads (XEnORDS)<n> Register (m=0–1, n=0–1)

Offset: Xor Engine0XOR Channel0: 0x00060880, Xor Engine0XOR Channel1: 0x00060884, Xor Engine1XOR Channel0: 0x00060980, Xor Engine1XOR Channel1: 0x00060984

Offset Formula: 0x00060880+4 * n + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x0	Reserved
2:0	MaxNoOfOutstandingReads	RW 0x4	Maximum Number of Outstanding Reads This field defines the maximum number of outstanding reads of source buffers data that is generated in DMA,CRC and various XOR modes of operation. The legal values are: 1, 2, 3, 4.

Copyright © 2015 Marvell

A.25.3 XOR Engine Descriptor Registers

Table 1848: XOR Engine Next Descriptor Pointer (XEnNDPR)<n> Register (m=0–1, n=0–1)
Offset: Xor Engine0XOR Channel0: 0x00060A00, Xor Engine0XOR Channel1: 0x00060A04, Xor Engine1XOR Channel0: 0x00060B00, Xor Engine1XOR Channel1: 0x00060B04
Offset Formula: 0x00060A00+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:0	NextDescPtr	RW 0x0	XOR Engine Next Descriptor Address Pointer In XOR mode, bits[5:0] must be zero. In CRC/DMA mode, bits[4:0] must be zero. **NOTE:** The value 0x0 is reserved for the end-of-chain NULL indication. Descriptors must not be placed at address 0x0. The XOR Engine will ignore attempts to read a descriptor from that address.

Table 1849: XOR Engine Current Descriptor Pointer (XEnCDPR)<n> Register (m=0–1, n=0–1)
Offset: Xor Engine0XOR Channel0: 0x00060A10, Xor Engine0XOR Channel1: 0x00060A14, Xor Engine1XOR Channel0: 0x00060B10, Xor Engine1XOR Channel1: 0x00060B14
Offset Formula: 0x00060A10+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:0	CurrentDescPtr	RO 0x0	XOR Engine current descriptor address pointer. Points to the last descriptor that was fetched.

Table 1850: XOR Engine Byte Count (XEnBCR)<n> Register (m=0–1, n=0–1)
Offset: Xor Engine0XOR Channel0: 0x00060A20, Xor Engine0XOR Channel1: 0x00060A24, Xor Engine1XOR Channel0: 0x00060B20, Xor Engine1XOR Channel1: 0x00060B24
Offset Formula: 0x00060A20+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:0	ByteCnt	RO 0x0	Number of bytes left for the XOR Engine to execute the current descriptor operation. In XOR, DMA, and Memory Initialization modes: updated after every write action. In CRC mode: updated after every calculation operation.

Copyright © 2015 Marvell
Document Classification: Proprietary Information

A.25.4 XOR Engine Memory Initialization

Table 1851: XOR Engine Destination Pointer (XEnDPR0)<n> Register (m=0–1, n=0–1)
Offset: Xor Engine0XOR Channel0: 0x00060AB0, Xor Engine0XOR Channel1: 0x00060AB4, Xor Engine1XOR Channel0: 0x00060BB0, Xor Engine1XOR Channel1: 0x00060BB4
Offset Formula: 0x00060AB0+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:0	DstPtr	RW 0x0	Points to the target block of the Memory Initialization operations. **NOTE:** Valid only for Memory Initialization mode. **NOTE:**

Table 1852: XOR Engine Block Size (XEnBSR)<n> Register (m=0–1, n=0–1)
Offset: Xor Engine0XOR Channel0: 0x00060AC0, Xor Engine0XOR Channel1: 0x00060AC4, Xor Engine1XOR Channel0: 0x00060BC0, Xor Engine1XOR Channel1: 0x00060BC4
Offset Formula: 0x00060AC0+n*4 + 0x100*m : where m (0-1) represents Xor Engine, where n (0-1) represents XOR Channel

Bit	Field	Type/InitVal	Description
31:0	BlockSize	RW 0x0	Size of block in bytes for Memory Initialization operation. This field along with the XE0DPR or XE1DPR (Destination pointer) register, defines the target block for these operations. Minimum value: 128B Maximum value: 4GB The value 0x00000000 stands for 4-GB block size. The block must not cross 4-GB boundary. **NOTE:** Valid only for Memory Initialization mode. **NOTE:**

Table 1853: XOR Engine Timer Mode Control (XETMCR) Register (m=0–1)
Offset: Xor Engine0: 0x00060AD0, Xor Engine1: 0x00060BD0
Offset Formula: 0x00060AD0+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:13	Reserved	RO 0x0	Reserved
12:8	SectionSizeCtrl	RW 0x0	Section Size Control Specifies the section size for ECC timer mode operation. Actual section size (in bytes) = 2^<SectionSizeCtrl> Minimum Value: 7 (section size of 128B). Maximum Value: 31 (section size of 2 GB). Must be less than Block Size (XEBSR) Reserved values: 0-6 **NOTE:** Valid only in ECC Timer mode.
7:1	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 1853: XOR Engine Timer Mode Control (XETMCR) Register (m=0–1) (Continued)
Offset: Xor Engine0: 0x00060AD0, Xor Engine1: 0x00060BD0
Offset Formula: 0x00060AD0+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
0	TimerEn	RW 0x0	Enable Timer Mode Enables triggering ECC operation with timer. If enabled the target block will be divided into sections according to the <SectionSizeCtrl> field value and the ECC timer will be enabled. Upon expiration of the timer, one section will be processed. When the timer expires the second time, the next section will be processed, and so on, until all the target blocks are processed. The XOR Engine will then again start cleaning the target block. To stop the operation, <XEstop> must be set. The ECC timer will be activated upon setting <XEstart> of a channel that is in ECC Timer mode. **NOTE:** Valid only in ECC mode. 0 = Disable 1 = Enable

Table 1854: XOR Engine Timer Mode Initial Value (XETMIVR) Register (m=0–1)
Offset: Xor Engine0: 0x00060AD4, Xor Engine1: 0x00060BD4
Offset Formula: 0x00060AD4+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:0	TimerInitVal	RW 0x0	ECC Timer Mode Initial Value for Count-Down Controls the time period between executions of subsequent sections ECC cleanup. Specifies the number of TCLK cycles between subsequent sections cleanup. If timer expires before current section cleanup has ended, it will be disregarded. **NOTE:** Valid only if one of the XOR Engine channels is in ECC timer mode.

Table 1855: XOR Engine Timer Mode Current Value (XETMCVR) Register (m=0–1)
Offset: Xor Engine0: 0x00060AD8, Xor Engine1: 0x00060BD8
Offset Formula: 0x00060AD8+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:0	TimerCrntVal	RO 0x0	ECC Timer Mode Current Value **NOTE:** Valid only if one of the XOR Engine channels is in ECC Timer mode.

Copyright © 2015 Marvell

Table 1856: XOR Engine Initial Value Low (XEIVRL) Register (m=0–1)
Offset: Xor Engine0: 0x00060AE0, Xor Engine1: 0x00060BE0
Offset Formula: 0x00060AE0+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:0	InitValL	RW 0x0	LSB of the initial value to be written cyclically to the target block in Memory Initialization mode. Mapped to bits[63:32] of the initial value. This register is shared between the 2 XOR Engine channels. The XOR Engine will compose a 64-bit initial value out of the XOR Engine Initial Value Low and XOR Engine Initial Value High registers and write it cyclically to the target block. The target block can be of any alignment. **NOTE:** Valid only on Memory Initialization modes.

Table 1857: XOR Engine Initial Value High (XEIVRH) Register (m=0–1)
Offset: Xor Engine0: 0x00060AE4, Xor Engine1: 0x00060BE4
Offset Formula: 0x00060AE4+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:0	InitValH	RW 0x0	MSB of the initial value to be written cyclically to the target block in Memory Initialization mode. Mapped to bits[63:32] of the initial value. This register is shared between the 2 XOR Engine channels. The XOR Engine will compose a 64-bit initial value out of the XOR Engine Initial Value Low and XOR Engine Initial Value High registers and write it cyclically to the target block. The target block can be of any alignment. **NOTE:** Valid only on Memory Initialization modes.

A.25.5 XOR Engine Interrupt Registers

Table 1858: XOR Engine Interrupt Cause (XEICR1) Register (m=0–1)
Offset: Xor Engine0: 0x00060830, Xor Engine1: 0x00060930
Offset Formula: 0x00060830+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
NOTE: All cause bits are clear only. They are set to 1 upon an interrupt event and cleared when the software writes a value of 0. Writing 1 has no effect (don't care). The XOR Engine disregards such write attempts.			
31:26	Reserved	RO 0x0	Reserved
25	XbarErr1	RW0C 0x0	Channel#1 Mbus Parity Error Caused by erroneous read response from the Mbus.
24	IntParityErr1	RW0C 0x0	Channel#1 Parity Error Caused by erroneous internal buffer read.
23	OwnErr1	RW0C 0x0	Channel#1 Descriptor Ownership Violation Attempt to access the descriptor owned by the CPU.
22	WrProt1	RW0C 0x0	Channel#1 Write Protect Violation Attempt to write to a window that is write protected.

Copyright © 2015 Marvell

Table 1858: XOR Engine Interrupt Cause (XEICR1) Register (m=0–1) (Continued)
Offset: Xor Engine0: 0x00060830, Xor Engine1: 0x00060930
Offset Formula: 0x00060830+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
21	AccProt1	RW0C 0x0	Channel#1 Access Protect Violation Attempt to access an address in a window in which access is not allowed.
20	AddrDecode1	RW0C 0x0	Channel#1 Failed address decoding Address is not in any window or matches more than one window.
19	Paused1	RW0C 0x0	Channel#1 Paused XOR Engine completed pausing routine, after receiving pause command (setting <XEpause>). It has entered Pause state.
18	Stopped1	RW0C 0x0	Channel#1 Stopped XOR Engine completed stopping routine, after receiving stop command (setting <XEstop>). It has entered Inactive state.
17	EOC1	RW0C 0x0	Channel#1 End of Chain Asserted when the XOR Engine finished transfer of current descriptor operation, and it is currently the last in the descriptor chain (byteCount==0) and (XENDP=NULL). Also asserted upon end of chain processing due to error condition.
16	EOD1	RW0C 0x0	Channel#1 End of Descriptor Asserted when the XOR Engine finished the transfer of the current descriptor operation (byteCount==0). Software can control EOD interrupt assertion per descriptor through the <EODIntEn> bit in the Command field of the descriptor.
15:10	Reserved	RO 0x0	Reserved
9	XbarErr0	RW0C 0x0	Channel#0 Mbus Parity Error Caused by an erroneous read response from the Mbus.
8	IntParityErr0	RW0C 0x0	Channel#0 Parity Error Caused by an erroneous internal buffer read.
7	OwnErr0	RW0C 0x0	Channel#0 Descriptor Ownership Violation Attempt to access the descriptor owned by the CPU.
6	WrProt0	RW0C 0x0	Channel#0 Write Protect Violation Attempt to write to a window that is write protected.
5	AccProt0	RW0C 0x0	Channel#0 Access Protect Violation Attempt to access an address in a window in which access is not allowed.
4	AddrDecode0	RW0C 0x0	Channel#0 Failed address decoding Address is not in any window or matches more than one window.
3	Paused0	RW0C 0x0	Channel#0 Paused XOR Engine completed pausing routine, after receiving pause command (setting <XEpause>). It has entered Pause state.
2	Stopped0	RW0C 0x0	Channel#0 Stopped XOR Engine completed stopping routine, after receiving stop command (setting <XEstop>). It has entered Inactive state.

Copyright © 2015 Marvell

Table 1858: XOR Engine Interrupt Cause (XEICR1) Register (m=0–1) (Continued)
Offset: Xor Engine0: 0x00060830, Xor Engine1: 0x00060930
Offset Formula: 0x00060830+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
21	AccProt1	RW0C 0x0	Channel#1 Access Protect Violation Attempt to access an address in a window in which access is not allowed.
20	AddrDecode1	RW0C 0x0	Channel#1 Failed address decoding Address is not in any window or matches more than one window.
19	Paused1	RW0C 0x0	Channel#1 Paused XOR Engine completed pausing routine, after receiving pause command (setting <XEpause>). It has entered Pause state.
18	Stopped1	RW0C 0x0	Channel#1 Stopped XOR Engine completed stopping routine, after receiving stop command (setting <XEstop>). It has entered Inactive state.
17	EOC1	RW0C 0x0	Channel#1 End of Chain Asserted when the XOR Engine finished transfer of current descriptor operation, and it is currently the last in the descriptor chain (byteCount==0) and (XENDP=NULL). Also asserted upon end of chain processing due to error condition.
16	EOD1	RW0C 0x0	Channel#1 End of Descriptor Asserted when the XOR Engine finished the transfer of the current descriptor operation (byteCount==0). Software can control EOD interrupt assertion per descriptor through the <EODIntEn> bit in the Command field of the descriptor.
15:10	Reserved	RO 0x0	Reserved
9	XbarErr0	RW0C 0x0	Channel#0 Mbus Parity Error Caused by an erroneous read response from the Mbus.
8	IntParityErr0	RW0C 0x0	Channel#0 Parity Error Caused by an erroneous internal buffer read.
7	OwnErr0	RW0C 0x0	Channel#0 Descriptor Ownership Violation Attempt to access the descriptor owned by the CPU.
6	WrProt0	RW0C 0x0	Channel#0 Write Protect Violation Attempt to write to a window that is write protected.
5	AccProt0	RW0C 0x0	Channel#0 Access Protect Violation Attempt to access an address in a window in which access is not allowed.
4	AddrDecode0	RW0C 0x0	Channel#0 Failed address decoding Address is not in any window or matches more than one window.
3	Paused0	RW0C 0x0	Channel#0 Paused XOR Engine completed pausing routine, after receiving pause command (setting <XEpause>). It has entered Pause state.
2	Stopped0	RW0C 0x0	Channel#0 Stopped XOR Engine completed stopping routine, after receiving stop command (setting <XEstop>). It has entered Inactive state.

Copyright © 2015 Marvell

Table 1858: XOR Engine Interrupt Cause (XEICR1) Register (m=0–1) (Continued)
Offset: Xor Engine0: 0x00060830, Xor Engine1: 0x00060930
Offset Formula: 0x00060830+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
1	EOC0	RW0C 0x0	Channel#0 End of Chain Asserted when the XOR Engine finished transfer of current descriptor operation, and it is currently the last in the descriptor chain (byteCount==0) and (XENDP=NULL). Also asserted upon end of chain processing due to error condition.
0	EOD0	RW0C 0x0	Channel#0 End of Descriptor Asserted when the XOR Engine finished the transfer of the current descriptor operation (byteCount==0). Software can control EOD interrupt assertion per descriptor through the <EODIntEn> bit in the Command field of the descriptor.

Table 1859: XOR Engine Interrupt Mask (XEIMR) Register (m=0–1)
Offset: Xor Engine0: 0x00060840, Xor Engine1: 0x00060940
Offset Formula: 0x00060840+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:26	Reserved	RO 0x0	Reserved
25:16	Channel1IntrptMask	RW 0x0	Channel1 Interrupt Mask Each bit in this field masks the interrupt generation due to the corresponding bit in the XOR Engine Interrupt Cause Register. 0 = Disable: Interrupt generation is disabled. 1 = Enable: Interrupt generation is enabled.
15:10	Reserved	RW 0x0	Reserved
9:0	Channel0IntrptMask	RW 0x0	Channel0 Interrupt Mask Each bit in this field masks the interrupt generation due to the corresponding bit in the XOR Engine Interrupt Cause Register. 0 = Disable: Interrupt generation is disabled. 1 = Enable: Interrupt generation is enabled.

Table 1860: XOR Engine Error Cause (XEECR) Register (m=0–1)
Offset: Xor Engine0: 0x00060850, Xor Engine1: 0x00060950
Offset Formula: 0x00060850+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:5	Reserved	RO 0x0	Reserved

Copyright © 2015 Marvell

Table 1860: XOR Engine Error Cause (XEECR) Register (m=0–1) (Continued)
Offset: Xor Engine0: 0x00060850, Xor Engine1: 0x00060950
Offset Formula: 0x00060850+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
4:0	ErrorType	ROC 0x0	Specifies the error event currently reported in the Error Address register: 0x0 = Null 0x1-0x3 = Reserved 0x4 = AddrDecode0 0x5 = AccProt0 0x6 = WrProt0 0x7-0x13 = Reserved 0x14 = AddrDecode1 0x15 = AccProt1 0x16 = WrProt1 0x17-0x1F = Reserved This field is self cleared by reading the Error Address Register (XEEAR). Once the error cause is latched, no new error cause or address is latched to the XOR Engine Error Cause (XEECR) or Error Address Register (XEEAR), until software reads the XEEAR register. Software should read XEECR first, and then read XEEAR.

Table 1861: XOR Engine Error Address (XEEAR) Register (m=0–1)
Offset: Xor Engine0: 0x00060860, Xor Engine1: 0x00060960
Offset Formula: 0x00060860+0x100*m: where m (0-1) represents Xor Engine

Bit	Field	Type/InitVal	Description
31:0	ErrAddr	RO 0x0	Bits[31:0] of Error Address Latched upon a violation event (address miss, multiple hit, access protect, write protect) on any of the address windows. Once the address is latched, no new address is latched until software reads it (Read access to the XOR Engine Error Address Register (XEEAR). SW should read the XOR Engine Error Cause Register (XEECR) first, and then the XOR Engine Error Address Register (XEEAR).

Copyright © 2015 Marvell

A.26 Multi-Purpose Ports (MPP) Registers

The following table provides a summarized list of all registers that belong to the Multi-Purpose Ports (MPP), including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1862: Summary Map Table for the Multi-Purpose Ports (MPP) Registers

Register Name	Offset	Table and Page
MPP_Control_<n*8>_<n*8+7> Register (n=0–6)	MPP_Num0: 0x00018000, MPP_Num1: 0x00018004, MPP_Num2: 0x00018008, MPP_Num3: 0x0001800C, MPP_Num4: 0x00018010, MPP_Num5: 0x00018014, MPP_Num6: 0x00018018	Table 1863, p. 1382
MPP Control 56_59 Register	0x0001801C	Table 1864, p. 1383

Table 1863: MPP_Control_<n*8>_<n*8+7> Register (n=0–6)
Offset: MPP_Num0: 0x00018000, MPP_Num1: 0x00018004, MPP_Num2: 0x00018008, MPP_Num3: 0x0001800C, MPP_Num4: 0x00018010, MPP_Num5: 0x00018014, MPP_Num6: 0x00018018
Offset Formula: 0x00018000+n*4: where n (0-6) represents MPP_Num

Bit	Field	Type/InitVal	Description
31:28	MPPSel<n*8+7>	RW 0x0	MPP<n*8+7> Select See the MPP Function Summary table in Hardware Specifications for this device.
27:24	MPPSel<n*8+6>	RW 0x0	MPP<n*8+6> Select See the MPP Function Summary table in Hardware Specifications for this device.
23:20	MPPSel<n*8+5>	RW 0x0	MPP<n*8+5> Select See the MPP Function Summary table in Hardware Specifications for this device.
19:16	MPPSel<n*8+4>	RW 0x0	MPP<n*8+4> Select See the MPP Function Summary table in Hardware Specifications for this device.
15:12	MPPSel<n*8+3>	RW 0x0	MPP<n*8+3> Select See the MPP Function Summary table in Hardware Specifications for this device.
11:8	MPPSel<n*8+2>	RW 0x0	MPP<n*8+2> Select See the MPP Function Summary table in Hardware Specifications for this device.

Copyright © 2015 Marvell

Table 1863: MPP_Control_<n*8>_<n*8+7> Register (n=0–6) (Continued)
Offset: MPP_Num0: 0x00018000, MPP_Num1: 0x00018004, MPP_Num2: 0x00018008, MPP_Num3: 0x0001800C, MPP_Num4: 0x00018010, MPP_Num5: 0x00018014, MPP_Num6: 0x00018018
Offset Formula: 0x00018000+n*4: where n (0-6) represents MPP_Num

Bit	Field	Type/InitVal	Description
7:4	MPPSel<n*8+1>	RW 0x0	MPP<n*8+1> Select See the MPP Function Summary table in Hardware Specifications for this device.
3:0	MPPSel<n*8>	RW 0x0	MPP<n*8> Select See the MPP Function Summary table in Hardware Specifications for this device.

Table 1864: MPP Control 56_59 Register
Offset: 0x0001801C

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	MPPSel59	RW 0x0	MPP59 Select See the MPP Function Summary table in Hardware Specifications for this device.
11:8	MPPSel58	RW 0x0	MPP58 Select See the MPP Function Summary table in Hardware Specifications for this device.
7:4	MPPSel57	RW 0x0	MPP57 Select See the MPP Function Summary table in Hardware Specifications for this device.
3:0	MPPSel56	RW 0x0	MPP56 Select See the MPP Function Summary table in Hardware Specifications for this device.

Copyright © 2015 Marvell

A.27 General Management - DFX Server Registers

The following table provides a summarized list of all registers that belong to the General Management - DFX Server, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1865: Summary Map Table for the General Management – DFX Server Registers

Register Name	Offset	Table and Page
DFX Server		
Core Divider Clock Control 0 Register	0x000E4250	Table 1866, p. 1384
Core Divider Clock Control 1 Register	0x000E4254	Table 1867, p. 1385
Core Divider Clock Control 2 Register	0x000E4258	Table 1868, p. 1385
Core Divider Clock Control 3 Register	0x000E425C	Table 1869, p. 1385
CPU PLL Clock Divider Control 0 Register	0x000E4260	Table 1870, p. 1386
CPU PLL Clock Divider Control 1 Register	0x000E4264	Table 1871, p. 1387
DFX Server General		
Server eFuse Control Register	0x000E4008	Table 1872, p. 1387
Server Reset Control Register	0x000E400C	Table 1873, p. 1388
PCIe Skip Initialization Matrix Register	0x000E406C	Table 1874, p. 1388
Temperature Sensor 28nm Control LSB Register	0x000E4070	Table 1875, p. 1388
Temperature Sensor 28nm Control MSB Register	0x000E4074	Table 1876, p. 1389
Temperature Sensor Status Register	0x000E4078	Table 1877, p. 1390
AVS Debug Control Register	0x000E4124	Table 1878, p. 1390
AVS Enabled Control Register	0x000E4130	Table 1879, p. 1390
Device ID Status Register	0x000E4240	Table 1880, p. 1390
JTAG Device ID Status Register	0x000E4244	Table 1881, p. 1391
DDR-PLL Parameters Register	0x000E42F0	Table 1882, p. 1391

A.27.1 DFX Server

Table 1866: Core Divider Clock Control 0 Register
Offset: 0x000E4250

Bit	Field	Type/InitVal	Description
31:9	Reserved	RSVD 0x7F88	Reserved
8	core_clkdiv_reload_ratio	RW 0x0	Central request to switch clocks ratios. Must be asserted until acknowledge by re-assertion of clkdiv_ratio_stable indicating that clocks are back to life.

Document Classification: Proprietary Information
Copyright © 2015 Marvell
November 25, 2015, Preliminary

Table 1866: Core Divider Clock Control 0 Register (Continued)
Offset: 0x000E4250

Bit	Field	Type/InitVal	Description
7:0	core_clkdiv_reload_force	RW 0x0	Reload Ratio trigger qualifier. Per clock tree. Request to force new ratio regardless of relation to other clocks. FieldBit-0 -> Reserved. FieldBit-1 -> NandFlash-Domain. FieldBit-2 -> Reserved.. FieldBit-3 -> Reserved. FieldBit-4 -> Reserved. FieldBit-5 -> Reserved. FieldBit-6 -> Reserved. FieldBit-7 -> Reserved.

Table 1867: Core Divider Clock Control 1 Register
Offset: 0x000E4254

Bit	Field	Type/InitVal	Description
31:14	Reserved	RSVD 0x2014	Reserved
13:8	corepll_clkdiv_nand_ecc_clk_ratio	RW 0x4	Full Integer Ratio from PLL-Out frequency to NAND-ECC block. The default value of this clock is: 250 MHz.
7:0	Reserved	RSVD 0x0	Reserved

Table 1868: Core Divider Clock Control 2 Register
Offset: 0x000E4258

Bit	Field	Type/InitVal	Description
31:0	Reserved	RSVD 0x19000002	Reserved

Table 1869: Core Divider Clock Control 3 Register
Offset: 0x000E425C

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x4	Reserved

Copyright © 2015 Marvell

Table 1869: Core Divider Clock Control 3 Register (Continued)
Offset: 0x000E425C

Bit	Field	Type/InitVal	Description
15:8	corepll_clkdiv_ratio_half	RW 0x0	Half Integer Ratio from PLL-Out frequency per clock tree. FieldBit-0 -> Reserved. FieldBit-1 -> NandFlash-Domain. FieldBit-2 -> Reserved.. FieldBit-3 -> Reserved. FieldBit-4 -> Reserved. FieldBit-5 -> Reserved. FieldBit-6 -> Reserved. FieldBit-7 -> Reserved.
7:0	Reserved	RSVD 0x4	Reserved

Table 1870: CPU PLL Clock Divider Control 0 Register
Offset: 0x000E4260

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:8	cpu_clkdiv_reload_smooth	RW 0x0	Reload Ratio trigger qualifier. Per clock tree. Request to modify new ratio, at the next convenient point – where effecting clocks that are not modified will remain intact and all associated cross-masks will transit smoothly. If a modified clock affects others (MC on CPU), the modification request will infer a restart of the affected clocks as well. It is recommended to use the smooth request below. FieldBit-0 -> MC-Domain. FieldBit-1 -> DDR-Domain. FieldBit-2 -> Reserved. FieldBit-3 -> ATCLK-Domain. FieldBit-4 -> PCLK-Domain. FieldBit-5 -> Reserved. FieldBit-6 -> Reserved. FieldBit-7 -> Reserved.
7:0	cpupll_clkdiv_reload_force	RW 0x0	Reload Ratio trigger qualifier. Per clock tree. Request to force new ratio regardless of relation to other clocks. It is recommended to use the Smooth request rather than the reload force. FieldBit-0 -> Reserved. FieldBit-1 -> DDR-Domain. FieldBit-2 -> Reserved. FieldBit-3 -> ATCLK-Domain. FieldBit-4 -> PCLK-Domain. FieldBit-5 -> Reserved. FieldBit-6 -> Reserved. FieldBit-7 -> Reserved.

Copyright © 2015 Marvell

Table 1871: CPU PLL Clock Divider Control 1 Register
Offset: 0x000E4264

Bit	Field	Type/InitVal	Description
31:30	Reserved	RSVD 0x0	Reserved
29:24	cpupll_clkdiv_l2_func_ratio	RW SAR	CPU to L2 Cache Clocks Ratio
23:16	Reserved	RSVD 0xFF	Reserved
15:12	cpupll_clkdiv_ref_cnt_base	RW 0x1	Sampling & Stabilization time in ref_clk's. The value in use is this baseline +1. (1 up to 16 total). Most ratios need less than 1 ref_clk, however – for extreme cases more may be needed(for example 6:40 – where an iteration takes 240 PLL-Out clocks, at 2.4GHz it still requires about 2.5 ref-clks, and to be on the safe side, wait two iterations – leading to 5-6 ref-clk cycles).
11:9	Reserved	RSVD 0x0	Reserved
8	cpupll_clkdiv_reload_ratio	RW 0x0	Central Request to Switch Ratios. There is no need to assert at "start of day". It is implied by exiting PLL reset. It is needed for all later requests for ratio modifications. It must be asserted until it is acknowledge by re-assertion of ClkDiv_Ratio_Stable, indicating that the clocks are operating.
7:0	cpupll_clkdiv_reset_mask	RW 0x0	Per clock tree reset masking – in case a repeated reset is not wanted, but may be asserted due to PLL re-stabilization or other events. These masks must be default set to zero for enabling initial reset. Later they can be selectively asserted as needed. FieldBit-0 -> MC-Domain. FieldBit-1 -> DDR-Domain. FieldBit-2 -> Reserved. FieldBit-3 -> ATCLK-Domain. FieldBit-4 -> PCLK-Domain. FieldBit-5 -> Reserved. FieldBit-6 -> Reserved. FieldBit-7 -> Reserved.

A.27.2 DFX Server General

Table 1872: Server eFuse Control Register
Offset: 0x000E4008

Bit	Field	Type/InitVal	Description
31	eFuse Program Enable	RW 0x0	This field is used to enable LD/HD eFuse programming. 0 = Disable: Disable Programming of LD/HD eFuse. 1 = Enable: Enable Programming of LD/HD eFuse.
30:7	Reserved	RSVD 0x1FF258	Reserved

Copyright © 2015 Marvell

Table 1872: Server eFuse Control Register (Continued)
Offset: 0x000E4008

Bit	Field	Type/InitVal	Description
6	ID eFUSE Select	RW 0x0	This field controls which ID eFuse (ID0/ID1) can get R/W access. 0 = ID0: The chip ID0 eFuse is slected. 1 = ID1: The chip ID1 eFuse is selected.
5:0	Reserved	RSVD 0x0	Reserved

Table 1873: Server Reset Control Register
Offset: 0x000E400C

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0x3	Reserved
3	PCIe Soft Resetn Mask	RW 0x1	Mask the external PCIe soft reset input to the General Management server.
2:0	Reserved	RSVD 0x2	Reserved

Table 1874: PCIe Skip Initialization Matrix Register
Offset: 0x000E406C

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x0	Reserved
0	PCIe Soft Reset Skip Init	RW 0x0	Skip the PCIe reset. The value for the Skip Init if Reset Source (ext_soft_reset_) is asserted.

Table 1875: Temperature Sensor 28nm Control LSB Register
Offset: 0x000E4070

Bit	Field	Type/InitVal	Description
31	Vsen Divide En	RW 0x0	This field enables the VSEN internal divider. 0 = Disable: Disable internal divider. Vsen_in[3:0] is measured as is. The allowed input voltage range in this mode is 0.55V to 1V. 1 = Enable: Enable internal divider. Vsen_in[3:0] is internally divided according to <vsen_cfg_divide>.
30:0	Reserved	RSVD 0x1064044	Reserved

Copyright © 2015 Marvell

Table 1876: Temperature Sensor 28nm Control MSB Register
Offset: 0x000E4074

Bit	Field	Type/InitVal	Description
31	Reserved	RSVD 0x0	Reserved
30:28	Temperature Sensor Unit Control	RW 0x6	This field defines which of the 5 TEMP Sensors on the chip are read in the server. 0 = 1ST: Read values of TEMP Sensor Number 0. 1 = 2ND: Read values of TEMP Sensor Number 1. 2 = 3RD: Read values of TEMP Sensor Number 2. 3 = 4TH: Read values of TEMP Sensor Number 3. 4 = 5TH: Read values of TEMP Sensor Number 4. 6 = MAX: Read values of hottest TEMP sensor. The TEMP sensor index appears on bits 31:29. 7 = AVG: Average value of all TEMP sensor. The number of averaged TEMP sensors appears on bits 31:29.
27:12	Reserved	RSVD 0x3414	Reserved
11:9	Tsen Sel Cfg Int Vref	RW 0x4	This field defines internal bias for trimming internal slope gen vref.
8	Tsen Reset	RW 0x0	This field initiates TSEN hardware reset (Active Low). 0 = Hardware reset 1 = Normal operation: Normal operation
7	Tsen Soft Reset	RW 0x0	This field initiates software reset (Active high). 0 = Normal operation 1 = Software reset
6	Tsen Power Down	RW 0x0	This field enables TSEN power down. **NOTE:** Getting out of power down with software reset only. 0 = Disable: Normal Operation mode. 1 = Enable: Power Down.
5:3	Reserved	RSVD 0x3	Reserved
2:0	Tsen Readout Avg	RW 0x1	This field defines the number of temperature reads that are averaged in the sensor readout. 0 = None: No Average 1 = 2PWR1: 2 temperature reads average 2 = 2PWR2: 4 temperature reads average 3 = 2PWR3: 8 temperature reads average 4 = 2PWR4: 16 temperature reads average 5 = 2PWR5: 32 temperature reads average

Copyright © 2015 Marvell

Table 1877: Temperature Sensor Status Register
Offset: 0x000E4078

Bit	Field	Type/InitVal	Description
31:11	Reserved	RSVD 0x0	Reserved
10	Temperature Sensor Readout is Valid	RO 0x0	This field indicates if the Temperature Sensor Readout field is valid. 0 = Not Valid: Temperature reading is not valid. 1 = Valid: Temperature reading is valid.
9:0	Temperature Sensor Readout	RO 0x0	This field shows the Temperature readout. Tj = 0.4761*ReadOut - 279.1

Table 1878: AVS Debug Control Register
Offset: 0x000E4124

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x100100E	Reserved
2:0	AVS Band-Gap Config	RW 0x4	This field defines the temperature curves of the Band-Gap circuit for AVS.

Table 1879: AVS Enabled Control Register
Offset: 0x000E4130

Bit	Field	Type/InitVal	Description
31:20	Reserved	RSVD 0x0	Reserved
19:12	AVS High VDD Limit	RW SAR	This field set the upper limit of AVS VDD. VDD is limited to (1.05/0.9)*(640 + FieldValue*10). **NOTE:** Maximum upper limit of VDD is 1341.7 mV. Bits [7:6] of this field are reserved.
11:4	AVS Low VDD Limit	RW SAR	This field set the lower limit of AVS VDD. VDD is limited to (1.05/0.9)*(640 + FieldValue*10). **NOTE:** Maximum lower limit of VDD is 746.7 mV. Bits [7:6] of this field are reserved.
3:0	Reserved	RSVD 0x4	Reserved

Table 1880: Device ID Status Register
Offset: 0x000E4240

Bit	Field	Type/InitVal	Description
31:20	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1880: Device ID Status Register (Continued)
Offset: 0x000E4240

Bit	Field	Type/InitVal	Description
19:16	Sw Revision Status	RO 0x0	Device Revision field of the PCIe DevID register
15:0	Device ID Status	RO 0x0	Device-ID number field of the PCIe DevID register

Table 1881: JTAG Device ID Status Register
Offset: 0x000E4244

Bit	Field	Type/InitVal	Description
31:28	JTAG Revision	RO 0x0	Revision number field of the JTAG IDCODE register
27:12	JTAG Device ID Status	RO 0x0	Device-ID number field of the JTAG IDCODE register
11:1	JTAG Manufacturer-ID	RO SAR	Manufactorer-ID number field of the JTAG IDCODE register
0	JTAG LSB	RO 0x1	LSB field of the JTAG IDCODE register - Shifted out first, always 1

Table 1882: DDR-PLL Parameters Register
Offset: 0x000E42F0

Bit	Field	Type/InitVal	Description
31:25	Reserved	RSVD 0x0	Reserved
24:21	DDR-PLL VCO BAND	RW SAR	DDR-PLL VCO BAND. Defines frequency range of PLL VCO. 2 = 2000-2600 MHz 4 = 3000-3400 MHz 7 = 3800-4000 MHz
20:12	DDR-PLL Ndiv	RW SAR	N divider value. N_div = N-1
11:3	DDR-PLL Mdiv	RW SAR	M divider value. M_div = M-1
2:0	DDR-PLL Kdiv	RW SAR	K divider value. **NOTE:** PLL freq. calculation formula is Fout = (Fin*N)/(M*K) 3 = 4 4 = 8 5 = 16

Copyright © 2015 Marvell

A.28 Clock Complex Registers

The following table provides a summarized list of all registers that belong to the Clock Complex, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1883: Summary Map Table for the Clock Complex Registers

Register Name	Offset	Table and Page
CPU Divider Clock Control Ratio Register	0x00018704	Table 1884, p. 1392
TDM DCO Divider Clock Control Register	0x00018730	Table 1885, p. 1392

Table 1884: CPU Divider Clock Control Ratio Register
Offset: 0x00018704

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x0	Reserved
5:0	Cpu Cpu0 ClkDiv Ratio Full	RW 0x2	Indicates when PMU overrides the CPU PLL configuration. Full Integer Ratio from PLL-Out frequency to CPU0 clock. **NOTE:** Controlled by Device-PMU during CPU-DFS flow.

Table 1885: TDM DCO Divider Clock Control Register
Offset: 0x00018730

Bit	Field	Type/InitVal	Description
31:26	DCOClkDivRatio	RW 0x6	DCO Clock Divider Ratio This field provides the divider ration for the clock from the DCO to the TDM unit.
25	DCOClkDivClockEn	RW 0x1	DCO Divider Clock Enable 0 = Disabled: The clock from DCO clock divider is gated. 1 = Enabled: The clock from DCO clock divider is running.
24	DCOClkDivReloadEn	RW 0x0	TDM clock from DCO reload enable Change frequency flow: 1. Gate the divided clock by setting <DCOClkDivClockEn> to 'Disabled'. 2. Set required ratio. 3. Set this bit to 1. 4. Wait for 1 msec. 5. Set this bit to 0. 6. Wait for 1 msec. 7. Enable the divided clock by setting <DCOClkDivClockEn> to 'Enabled'.
23:0	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information
Copyright © 2015 Marvell

A.29 System Registers

The following table provides a summarized list of all registers that belong to the System, including the names, their type, offset, and a reference to the corresponding table and page for a detailed description of each element and its fields.

Table 1886: Summary Map Table for the System Registers

Register Name	Offset	Table and Page
Configuration and Control		
SoC Control 0 Register	0x00018200	Table 1887, p. 1394
SoC Control 1 Register	0x00018204	Table 1888, p. 1394
SoC Device Multiplex Register	0x00018208	Table 1889, p. 1396
Device ID Register	0x00018238	Table 1890, p. 1397
Device and Version ID Register	0x0001823C	Table 1891, p. 1397
SYSRSTn Length Counter Register	0x00018250	Table 1892, p. 1397
CPU Parity Fail Error Register	0x00018254	Table 1893, p. 1398
RSTOUTn Mask Register	0x00018260	Table 1894, p. 1399
System Soft Reset Register	0x00018264	Table 1895, p. 1400
Device Capabilities 0 Register	0x00018284	Table 1896, p. 1402
Device Capabilities 1 Register	0x00018288	Table 1897, p. 1402
PCI Express Boot Address Register	0x000182D4	Table 1898, p. 1403
General Purpose Reserved0 Register	0x000182E0	Table 1899, p. 1403
IP Configuration		
TDM PLL Configuration 0 Register	0x00018400	Table 1900, p. 1403
TDM PLL Configuration 1 Register	0x00018404	Table 1901, p. 1404
TDM PLL Configuration 2 Register	0x00018408	Table 1902, p. 1405
Audio PLL Configuration 0 Register	0x00018410	Table 1903, p. 1405
Audio PLL Configuration 1 Register	0x00018414	Table 1904, p. 1407
Audio PLL Configuration 2 Register	0x00018418	Table 1905, p. 1407
USB Configuration Register	0x00018420	Table 1906, p. 1408
UTMI PHY<n> Configuration Status Register (n=0–2)	PHY_NUM0: 0x00018440, PHY_NUM1: 0x00018444, PHY_NUM2: 0x00018448	Table 1907, p. 1408
SDIO3 Configuration Register	0x00018454	Table 1908, p. 1409
QSGMII Control 1 Register	0x00018494	Table 1909, p. 1409
QSGMII Status Register	0x00018498	Table 1910, p. 1410
RTC SoC Interrupt Register	0x000184A8	Table 1911, p. 1411
SSCG Frequency Configuration Register	0x000184AC	Table 1912, p. 1411
SSCG SSC Mode Register	0x000184B0	Table 1913, p. 1411
LDO Control Register	0x000184C0	Table 1914, p. 1412
IO Configuration 1 Register	0x000184E4	Table 1915, p. 1413
IO Configuration 2 Register	0x000184E8	Table 1916, p. 1413
IO Configuration 4 Register	0x000184F4	Table 1918, p. 1414
Power Management and Memory Power Down		

Copyright © 2015 Marvell

Table 1886: Summary Map Table for the System Registers (Continued)

Register Name	Offset	Table and Page
Power Management Clock Gating Control Register	0x00018220	Table 1919, p. 1414
BootROM		
BootROM Routine and Error Code Register	0x000182D0	Table 1920, p. 1416
Interrupt		
Misc Error Interrupt Cause Register	0x000182A8	Table 1921, p. 1418
Misc Error Interrupt Mask Register	0x000182AC	Table 1922, p. 1419
Design for Testablity (DFT)		
Sample at Reset (SAR) Register	0x00018600	Table 1923, p. 1420
PUP Enable Register	0x0001864C	Table 1924, p. 1422
On-Die Pattern Generator (ODPG) Enable Register	0x000186D4	Table 1925, p. 1422

A.29.1 Configuration and Control

Table 1887: SoC Control 0 Register
Offset: 0x00018200

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27	RefClkOut40M_1	RW 0x0	Selects the source clock to REF_CLK_OUT[1] in the MPP. 0 = 25 MHz 1 = 40 MHz
26:20	Reserved	RSVD 0x0	Reserved
19	RefClkOut40M_0	RW 0x0	Selects the source clock to REF_CLK_OUT[0] in the MPP. 0 = 25 MHz 1 = 40 MHz
18:0	Reserved	RSVD 0x14EE7	Reserved

Table 1888: SoC Control 1 Register
Offset: 0x00018204

Bit	Field	Type/InitVal	Description
31	Reserved	RW 0x0	Reserved Must be 0.
30:28	Reserved	RSVD 0x0	Reserved

Document Classification: Proprietary Information

Table 1888: SoC Control 1 Register (Continued)
Offset: 0x00018204

Bit	Field	Type/InitVal	Description
27	AudioMode	RW 0x0	Selects the Audio Unit Mode 0 = I2S 1 = SPDIF
26:16	Reserved	RSVD 0x707	Reserved
15	PCIe0 Intr Mask	RW 0x1	Mask a request from the CPU to the PCIe to generate MSI or interrupt message. When PCIe port0 is configured as EP, this port must be cleared to 0. 0 = Disable 1 = Enable
14	PCIe0QuadBy4En	RW 0x1	PCI Express ports configuration. Changing this field must be done at system initialization and before the assertion of PCIe0En. To access PCIe1, PCIe2, and PCIe3 from any agent the value of this field must be 0x1 (Enable). 0 = Disable: PCIe0 maximum link-width is by 4. PCIe1, PCIe2, and PCIe3 are disconnect. 1 = Enable: PCIe0, PCIe1, PCIe2, and PCIe3 are connected and their maximum link-width is by 1.
13:6	Reserved	RSVD 0x3	Reserved
5	PCIe1_CLKOUT_EN	RW 0x1	PCIe1 CLKOUT Enable 0 = Disable: Clock disabled. 1 = Enable: Clock enabled.
4	PCIe0_CLKOUT_EN	RW 0x1	PCIe0 CLKOUT Enable 0 = Disable: Clock disabled. 1 = Enable: Clock enabled.
3	PCIe3En	RW 0x0	PCI Express Port3 Enable **NOTE:** If PCIe1QuadBy1En is QuadX1 then this field is relevant for all the 4*PCIe_X1. 0 = Disable 1 = Enable
2	PCIe2En	RW 0x0	PCI Express Port2 Enable **NOTE:** If PCIe1QuadBy1En is QuadX1 then this field is relevant for all the 4*PCIe_X1. 0 = Disable 1 = Enable
1	PCIe1En	RW 0x0	PCI Express Port1 Enable **NOTE:** If PCIe1QuadBy1En is QuadX1 then this field is relevant for all the 4*PCIe_X1. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1888: SoC Control 1 Register (Continued)
Offset: 0x00018204

Bit	Field	Type/InitVal	Description
0	PCIe0En	RW 0x0	PCI Express Port0.X Enable **NOTE:** If PCIe0QuadBy1En is QuadX1 then this field is relevant for all the 4*PCIe_X1. 0 = Disable 1 = Enable

Table 1889: SoC Device Multiplex Register
Offset: 0x00018208

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27	NfArbiterEn	RW 0x0	Enables arbitration between device and NAND flash.
26	NfExpandTrrTimingEn	RW 0x0	Controls minimal tRR delay time (NF_REn de-assertion to NF_RDYn assertion). 0 = Minimal Delay: (4-5 Clock cycles) 1 = Expand Minimal Delay: (4-5 Clock cycles) by additional 5 Clock cycles.
25	NfCtrlIntMask	RW 0x0	NAND Flash Controller Interrupt Mask Register 0 = Interrupt mask 1 = Interrupt enable
24	NfCtrlIntCause	RW0C 0x0	NAND Flash Controller Interrupt Cause Register
23	Reserved	RSVD 0x0	Reserved
22	NfClkInvert	RW 0x0	NAND Clock Polarity 0 = Normal 1 = Inverted
21	NfCoreSwRst_	RW 0x1	NAND Core Clock Software Reset 0 = Enable 1 = Disable
20	NfEccSwRst_	RW 0x1	NAND ECC Clock Software Reset 0 = Enable 1 = Disable
19	NfAddressMaskEn	RW 0x0	NAND External Access Enable 0 = Disable: Internal access only (for 4B PIO). 1 = Enable: Enables external access (for 8B PIO).
18	NfCsExpansionEn	RW 0x0	Chip Select Expansion Enable 0 = Disable 1 = Enable
17	NfWrErrPropEn	RW 0x0	Write Response Error Propagate Enable 0 = Disable 1 = Enable
16:1	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1889: SoC Device Multiplex Register (Continued)
Offset: 0x00018208

Bit	Field	Type/InitVal	Description
0	NandFlashEnabled	RW SAR	Configure NAND Flash Enable 0 = Disable: NAND Flash Controller is disabled. 1 = Enable: NAND Flash Controller is enabled.

Table 1890: Device ID Register
Offset: 0x00018238

Bit	Field	Type/InitVal	Description
31:16	DevID	RW 0x6920	Device ID
15:0	VenID	RO 0x11AB	Vendor ID This field identifies Marvell as the Vendor of the device.

Table 1891: Device and Version ID Register
Offset: 0x0001823C

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x0	Reserved
11:8	Version ID	RO 0x0	See the Hardware Specifications for more details.
7:0	Reserved	RSVD 0x0	Reserved

Table 1892: SYSRSTn Length Counter Register
Offset: 0x00018250

Bit	Field	Type/InitVal	Description
31	Clr	WO 0x0	Writing 0x1 to this field clears the <Count> field.
30:29	Reserved	RSVD 0x0	Reserved
28:0	Count	RO 0x0	SYSRST Length Count. The value in this field specifies the number of REF_CLK cycles that the SYSRSTn was asserted. Software may clear this field by writing 0x1 to <Clr>.

Copyright © 2015 Marvell

Table 1893: CPU Parity Fail Error Register
Offset: 0x00018254

Bit	Field	Type/InitVal	Description
31:24	CPU1 Parity Fail Mask	RW 0xff	Mask bit for each CPU1 parity fail bit
23:16	CPU0 Pairy Fail Mask	RW 0xff	Mask bit for each CPU0 parity fail bit
15	CPU1 Parity Fail 7	RW0C 0x0	BTAC parity error CPU1
14	CPU1 Parity Fail 6	RW0C 0x0	GHB parity error CPU1
13	CPU1 Parity Fail 5	RW0C 0x0	Instruction tag RAM parity error CPU1
12	CPU1 Parity Fail 4	RW0C 0x0	Instruction data RAM parity error CPU1
11	CPU1 Parity Fail 3	RW0C 0x0	Main TLB parity error CPU1
10	CPU1 Parity Fail 2	RW0C 0x0	D outer RAM parity error CPU1
9	CPU1 Parity Fail 1	RW0C 0x0	Data tag RAM parity error CPU1
8	CPU1 Parity Fail 0	RW0C 0x0	Data RAM parity error for CPU1
7	CPU0 Parity Fail 7	RW0C 0x0	BTAC parity error CPU0
6	CPU0 Parity Fail 6	RW0C 0x0	GHB parity error CPU0
5	CPU0 Parity Fail 5	RW0C 0x0	Instruction tag RAM parity error CPU0
4	CPU0 Parity Fail 4	RW0C 0x0	Instruction data RAM parity error CPU0
3	CPU0 Parity Fail 3	RW0C 0x0	Main TLB parity error CPU0
2	CPU0 Parity Fail 2	RW0C 0x0	D outer RAM parity error CPU0
1	CPU0 Parity Fail 1	RW0C 0x0	Data tag RAM parity error CPU0
0	CPU0 Parity Fail 0	RW0C 0x0	Data RAM parity error for CPU0

Copyright © 2015 Marvell

Table 1894: RSTOUTn Mask Register
Offset: 0x00018260

Bit	Field	Type/InitVal	Description
31:11	Reserved	RSVD 0x0	Reserved
10	WatchDogMaskSysRstOut	RW 0x1	Mask Watchdog Reset from influencing the SYSRST_OUTn 0 = Do Not Mask: Do not mask Watchdog from influencing the SYSRST_OUTn. 1 = Mask: Mask Watchdog from influencing the SYSRST_OUTn.
9	MRnMaskSysRstOut	RW 0x0	Mask MRn from influencing the SYSRST_OUTn 0 = Do Not Mask: Do not mask MRn from influencing the SYSRST_OUTn. 1 = Mask: Mask MRn from influencing the SYSRST_OUTn.
8	PCIe3RstOutMaskTrst	RW 0x1	Mask PCIe3_Rst_Out_ from influencing the Trst_Tree. 0 = Do Not Mask: Do not mask PCIe3_Rst_Out from influencing the Trst_Tree. 1 = Mask: Mask PCIe3_Rst_Out from influencing the Trst_Tree.
7	PCIe2RstOutMaskTrst	RW 0x1	Mask PCIe2_Rst_Out_ from influencing the Trst_Tree. 0 = Do Not Mask: Do not mask PCIe2_Rst_Out from influencing the Trst_Tree. 1 = Mask: Mask PCIe2_Rst_Out from influencing the Trst_Tree.
6	PCIe1RstOutMaskTrst	RW 0x1	Mask PCIe1_Rst_Out_ from influencing the Trst_Tree. 0 = Do Not Mask: Do not mask PCIe1_Rst_Out from influencing the Trst_Tree. 1 = Mask: Mask PCIe1_Rst_Out from influencing the Trst_Tree.
5	PCIe0RstOutMaskTrst	RW 0x1	Mask PCIe0_Rst_Out_ from influencing the Trst_Tree. 0 = Do Not Mask: Do not mask PCIe0_Rst_Out from influencing the Trst_Tree. 1 = Mask: Mask PCIe0_Rst_Out from influencing the Trst_Tree.
4	PCIe3RstOutMaskSysRstOut	RW 0x1	Mask PCIe3_Rst_Out from influencing the SYSRST_OUTn 0 = Do Not Mask: Do not mask PCIe3_Rst_Out from influencing the SYSRST_OUTn. 1 = Mask: Mask PCIe3_Rst_Out from influencing the SYSRST_OUTn.
3	PCIe2RstOutMaskSysRstOut	RW 0x1	Mask PCIe2_Rst_Out from influencing the SYSRST_OUTn 0 = Do Not Mask: Do not mask PCIe2_Rst_Out from influencing the SYSRST_OUTn. 1 = Mask: Mask PCIe2_Rst_Out from influencing the SYSRST_OUTn.
2	PCIe1RstOutMaskSysRstOut	RW 0x1	Mask PCIe1_Rst_Out from influencing the SYSRST_OUTn 0 = Do Not Mask: Do not mask PCIe1_Rst_Out from influencing the SYSRST_OUTn. 1 = Mask: Mask PCIe1_Rst_Out from influencing the SYSRST_OUTn.
1	PCIe0RstOutMaskSysRstOut	RW 0x1	Mask PCIe0_Rst_Out from influencing the SYSRST_OUTn 0 = Do Not Mask: Do not mask PCIe0_Rst_Out from influencing the SYSRST_OUTn. 1 = Mask: Mask PCIe0_Rst_Out from influencing the SYSRST_OUTn.

Copyright © 2015 Marvell

Table 1894: RSTOUTn Mask Register (Continued)
Offset: 0x00018260

Bit	Field	Type/InitVal	Description
0	GlobalSoftRstOutEn	RW 0x0	If set to 1, the device asserts RSTOUTn upon SW reset.

Table 1895: System Soft Reset Register
Offset: 0x00018264

Bit	Field	Type/InitVal	Description
31:28	Reserved	RSVD 0x0	Reserved
27	PNCSwFastRstn	RW 0x1	PNC Software Fast RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
26	PNCSwRstn	RW 0x1	PNC Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
25	EIP150FSwRstn	RW 0x0	EIP150F Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
24	Usb3DeviceSwRstn	RW 0x1	USB3 Device Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
23	Usb3Host1SwRstn	RW 0x1	USB3 Host 1 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
22	Usb3Host0SwRstn	RW 0x1	USB3 Host 0 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
21	Reserved	RSVD 0x1	Reserved
20	GbE12SwFastRstn	RW 0x1	GbE ports1 and 2 Software Fast RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
19	GbE0SwFastRstn	RW 0x1	GbE port 0 Software Fast RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
18	GbE12SwRstn	RW 0x1	GbE ports1 and 2 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
17	GbE0SwRstn	RW 0x1	GbE port 0 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
16	Sata1SwRstn	RW 0x1	SATA port 2 and SATA port 3 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.

Copyright © 2015 Marvell

Table 1895: System Soft Reset Register (Continued)
Offset: 0x00018264

Bit	Field	Type/InitVal	Description
15	Sata0SwRstn	RW 0x1	SATA port 0 and SATA port 1 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
14	PCIe0SwRstn	RW 0x1	PCIe port 0 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
13	PCIe3SwRstn	RW 0x1	PCIe port 3 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
12	PCIe2SwRstn	RW 0x1	PCIe port 2 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
11	PCIe1SwRstn	RW 0x1	PCIe port 1 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
10	Crypto1SwRstn	RW 0x1	Cryptographic engine 1 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
9	Crypto0SwRstn	RW 0x1	Cryptographic engine 0 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
8	XorSwRstn	RW 0x1	XOR engine Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
7	Usb2SwRstn	RW 0x1	USB2 Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
6	Reserved	RSVD 0x0	Reserved
5	AudioSwRstn	RW 0x1	Audio Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
4	SdioSwTrstn	RW 0x1	0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
3	SdioSwArstn	RW 0x1	SDIO Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
2	BmSwRstn	RW 0x1	Buffer Management Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
1	TDMSwRstn	RW 0x1	TDM Software RESETn 0x0 = Reset is asserted. 0x1 = Reset is de-asserted.
0	GlobalSoftRst	RW 0x0	When software sets this bit to 1 and bit <SoftRstOutEn> is set to 1 in the RSTOUTn Mask Register, the device asserts the RSTOUTn pin.

Copyright © 2015 Marvell

Table 1896: Device Capabilities 0 Register
Offset: 0x00018284

Bit	Field	Type/InitVal	Description
31:27	Reserved	RSVD 0x0	Reserved
26	Reserved	RW 0x1	Reserved
25	Reserved	RW 0x1	Reserved
24	Reserved	RW 0x1	Reserved
23	Reserved	RSVD 0x0	Reserved
22:16	Reserved	RW 0x7f	Reserved
15:14	Reserved	RW 0x3	Reserved
13:12	Reserved	RW 0x3	Reserved
11:8	Reserved	RW 0xf	Reserved
7:4	Reserved	RW 0xf	Reserved
3:0	Reserved	RW 0xf	Reserved

Table 1897: Device Capabilities 1 Register
Offset: 0x00018288

Bit	Field	Type/InitVal	Description
31:2	Reserved	RSVD 0x0	Reserved
1	Reserved	RW 0x0	Reserved
0	Reserved	RW 0x0	Reserved

Copyright © 2015 Marvell

Table 1898: PCI Express Boot Address Register
Offset: 0x000182D4

Bit	Field	Type/InitVal	Description
31:0	BootAddr	RW 0xFFFF0000	The 32-bit address of the bootROM header. After initializing the PCI Express to operate as an Endpoint, the bootROM sets this register to 0xFFFFFFFF and waits for the register to be updated by the Root Complex.

Table 1899: General Purpose Reserved0 Register
Offset: 0x000182E0

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x1FFFE000	Reserved
2	CONF_SATA_LED_MODE	RW 0x0	Select the MPP LED mode activation. 0 = Present_act: Present and Active signal effect on LED. Only when present and not active, LED will be on. 1 = Act_only: Active connected directly. When not active, LED is on.
1:0	Reserved	RSVD 0x0	Reserved

A.29.2 IP Configuration

Table 1900: TDM PLL Configuration 0 Register
Offset: 0x00018400

Bit	Field	Type/InitVal	Description
31	TDM Apll Pi En	RW 0x1	Enables Signal for Interpolator Block. 0 = Interpolator disabled 1 = Interpolator enabled
30	TDM Apll Reset Intp Ext	RW 0x0	External Interpolator Reset Signal. 0 = No reset 1 = Reset
29	TDM Apll Clk Det En	RW 0x1	PI Output Clock Enable. Enables PI output clock for internal reset circuit. 0 = Disable 1 = Enable
28:27	TDM Apll Intpi	RW 0x2	Interpolator Bias Current Select. Selects Interpolator bias current based on the running frequency of PHASE[8:1]. 0 = 60 ~80 MHz 1 = 80 ~110 MHz 2 = 110 ~150 MHz 3 = 150 ~200 MHz

Copyright © 2015 Marvell

Table 1900: TDM PLL Configuration 0 Register (Continued)
Offset: 0x00018400

Bit	Field	Type/InitVal	Description
26:25	TDM Apll Fd Sel	RW 0x1	Lock Detector Accuracy Select.
24:23	TDM Apll Ctune	RW 0x1	VCO Capacitor Select. 0 = No Capacitor Loading 1 = 1 Unit Capacitor Loading 2 = 2 Unit Capacitor Loading 3 = 3 Unit Capacitor Loading
22:19	TDM Apll Icp	RW 0x5	Charge-pump Current Control. 0 = 1.5 µA 1 = 1.75 µA 2 = 2 µA 3 = 2.25 µA 4 = 3 µA 5 = 3.75 µA 6 = 4.5 µA 7 = 5.25 µA 8 = 6 µA 9 = 7.5 µA 10 = 9 µA 11 = 10.5 µA 12 = 12 µA 13 = 15 µA 14 = 18 µA 15 = 21 µA
18:10	TDM Apll Fbdiv	RW 0x1d	Feedback Clock Divider Select. Divider = FBDIV[8:0], start from 1.
9:1	TDM Apll Refdiv	RW 0x5	Reference Clock Divider Select. Divider = REFDIV[8:0], start from 1.
0	TDM Apll Pu	RW 0x1	Power-up signal for the PLL.

Table 1901: TDM PLL Configuration 1 Register
Offset: 0x00018404

Bit	Field	Type/InitVal	Description
31	TDM Apll Sw Reset	RW 0x1	Software reset for the TDM APLL 0 = Enable: Assert TDM APLL software reset. 1 = Disable: De-assert TDM APLL software reset.
30	TDM Apll Clkout Vco Div En	RW 0x0	CLKOUT_VCO_DIV Clock Control Signal. 0 = Disable: Disables CLKOUT_VCO_DIV clock. 1 = Enable: Enables CLKOUT_VCO_DIV.
29	TDM Apll Clkout Vco En	RW 0x0	CLKOUT_VCO Clock Control Signal. 0 = Disable: Disables CLKOUT_VCO clock 1 = Enable: Enables CLKOUT_VCO.
28:19	TDM Apll Reserve In	RW 0x0	Reserved Input Register Pin.

Copyright © 2015 Marvell

Table 1901: TDM PLL Configuration 1 Register (Continued)
Offset: 0x00018404

Bit	Field	Type/InitVal	Description
18	TDM Apll Postdiv Audio En	RW 0x1	External Offset Logic Reset Signal. 0 = No reset 1 = Reset
17	TDM Apll Reset Offset Ext	RW 0x0	Power-up Signal for the PLL.
16	TDM Apll Freq Offset Valid	RW 0x0	Frequency Offset Value Readiness Indicator. This is the indicator for FREQ_OFFSET[15:0]. The pulse width should be > 48/F VCO.
15:0	TDM Apll Freq Offset	RW 0x0	Frequency Offset Setting. [15]: Sign-Bit. 0 = Frequency down 1 = Frequency up [14:0]: 1 LSB -> 2ppm, up to 5%

Table 1902: TDM PLL Configuration 2 Register
Offset: 0x00018408

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:7	TDM Apll Clkout Vco Div Sel	RW 0x0	VCO Post Divider. 0x000 = 1 0x001 = 1 0x002 = 2 0x003 = 3 0x004 = 4 ... 0x1FF = 511.
6:0	TDM Apll Postdiv Audio	RW 0x3	Set Audio Post-Divider. Divider = 2, POSTDIV_AUDIO[6:0] = 2 Divider = 3, POSTDIV_AUDIO[6:0] = 3 Divider = 9, POSTDIV_AUDIO[6:0] = 9 Divider = 2*POSTDIV_AUDIO[6:1], start from POSTDIV_AUDIO[6:0] = 2 **NOTE:** Invalid setting for POSTDIV_AUDIO[5:0] = 0,1, and other odd number (except 3 and 9).

Table 1903: Audio PLL Configuration 0 Register
Offset: 0x00018410

Bit	Field	Type/InitVal	Description
31	Audio Apll Pi En	RW 0x1	Enable Signal For Interpolator Block. 0 = Disable: Interpolator is off. 1 = Enable: Interpolator is on.

Copyright © 2015 Marvell

Table 1903: Audio PLL Configuration 0 Register (Continued)
Offset: 0x00018410

Bit	Field	Type/InitVal	Description
30	Audio Apll Reset Intp Ext	RW 0x0	External Interpolator Reset Signal 0 = No reset 1 = Reset
29	Audio Apll Clk Det En	RW 0x1	PI Output Clock Enable. Enables PI output clock for internal reset circuit. 0 = Disable 1 = Enable
28:27	Audio Apll Intpi	RW 0x2	Interpolator Bias Current Select. Selects Interpolator bias current based on the running frequency of PHASE[8:1]. 0 = 60 ~80 MHz 1 = 80 ~110 MHz 2 = 110 ~150 MHz 3 = 150 ~200 MHz
26:25	Audio Apll Fd Sel	RW 0x1	Lock Detector Accuracy Select.
24:23	Audio Apll Ctune	RW 0x1	VCO Capacitor Select. 0 = No Capacitor Loading 1 = 1 Unit Capacitor Loading 2 = 2 Unit Capacitor Loading 3 = 3 Unit Capacitor Loading
22:19	Audio Apll Icp	RW 0x5	Charge-pump Current Control. 0 = 1.5 µA 1 = 1.75 µA 2 = 2 µA 3 = 2.25 µA 4 = 3 µA 5 = 3.75 µA 6 = 4.5 µA 7 = 5.25 µA 8 = 6 µA 9 = 7.5 µA 10 = 9 µA 11 = 10.5 µA 12 = 12 µA 13 = 15 µA 14 = 18 µA 15 = 21 µA
18:10	Audio Apll Fbdiv	RW 0x1d	Feedback Clock Divider Select. Divider = FBDIV[8:0], start from 1.
9:1	Audio Apll Refdiv	RW 0x5	Reference Clock Divider Select. Divider = REFDIV[8:0], start from 1.
0	Audio Apll Pu	RW 0x1	Power-up Signal for the PLL.

Copyright © 2015 Marvell

Table 1904: Audio PLL Configuration 1 Register
Offset: 0x00018414

Bit	Field	Type/InitVal	Description
31	Audio Apll Sw Reset_	RW 0x1	Software reset for the AUDIO APLL 0 = Enable: Assert Audio APLL software reset 1 = Disable: De-assert Audio APLL software reset
30	Audio Apll Clkout Vco Div En	RW 0x0	CLKOUT_VCO_DIV Clock Control Signal. 0 = Disable: Disables CLKOUT_VCO_DIV clock. 1 = Enable: Enables CLKOUT_VCO_DIV clock.
29	Audio Apll Clkout Vco En	RW 0x0	CLKOUT_VCO Clock Control Signal. 0 = Disable: Disables CLKOUT_VCO clock. 1 = Enable: Enables CLKOUT_VCO clock.
28:19	Audio Apll Reserved In	RW 0x0	Reserved Input Register Pin.
18	Audio Apll Postdiv Audio En	RW 0x1	External Offset Logic Reset Signal. 0 = No reset 1 = Reset
17	Audio Apll Reset Offset Ext	RW 0x0	Power-up Signal for the PLL.
16	Audio Apll Freq Offset Valid	RW 0x0	Frequency Offset Value Readiness Indicator. This is the indicator for FREQ_OFFSET[15:0]. The pulse width should be > 48/F VCO.
15:0	Audio Apll Freq Offset	RW 0x0	Frequency Offset Setting. [15]: Sign-Bit. 0 = Frequency down 1 = Frequency up [14:0]: 1 LSB -> 2ppm, up to 5%

Table 1905: Audio PLL Configuration 2 Register
Offset: 0x00018418

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:7	Audio Apll Clkout Vco Div Sel	RW 0x0	VCO Post Divider. 0x000 = 1 0x001 = 1 0x002 = 2 0x003 = 3 0x004 = 4 ... 0x1FF = 511.

Copyright © 2015 Marvell

Table 1905: Audio PLL Configuration 2 Register (Continued)
Offset: 0x00018418

Bit	Field	Type/InitVal	Description
6:0	Audio Apll Postdiv Audio	RW 0x3	Set Audio Post-Divider. Divider = 2, POSTDIV_AUDIO[6:0]= 2 Divider = 3, POSTDIV_AUDIO[6:0] = 3 Divider = 9, POSTDIV_AUDIO[6:0] = 9 Divider=2*POSTDIV_AUDIO[6:1], start from POSTDIV_AUDIO[6:0] = 2 **NOTE:** Invalid setting for POSTDIV_AUDIO[5:0] = 0,1, and other odd number (except 3 and 9).

Table 1906: USB Configuration Register
Offset: 0x00018420

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25	PU PLL	RW 0x1	UTMI Phy PU PLL 0 = Disable 1 = Enable
24	UTMI0ConfFromMAC	RW 0x1	UTMI0 Configured from MAC Enable 0 = Disable: UTMI0 configured from RFU. 1 = Enable: UTMI0 configured from USB2 MAC.
23:2	Reserved	RSVD 0x820	Reserved
1	USB3 Device UTMI Mux	RW 0x0	USB3 Device UTMI Mux 0 = UTMI2: USB3 Device is on UTMI 2 (if USB3 Device is enabled). 1 = UTMI0: USB3 Device is on UTMI 0 (if USB3 Device is enabled).
0	USB3 Device UTMI En	RW 0x0	USB3 Device UTMI Enable 0 = Disable: USB3 Device disabled. (USB2 on UTMI0, and USB3 Host1 on UTMI2). 1 = Enable: USB3 Device enabled.

Table 1907: UTMI PHY<n> Configuration Status Register (n=0–2)
Offset: PHY_NUM0: 0x00018440, PHY_NUM1: 0x00018444, PHY_NUM2: 0x00018448
Offset Formula: 0x00018440+n*4: where n (0-2) represents PHY_NUM

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x1	Reserved
5	PU PHY	RW 0x1	Power Up for the PHY 0 = Power Off 1 = Power On
4:2	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1907: UTMI PHY<n> Configuration Status Register (n=0–2) (Continued)
Offset: PHY_NUM0: 0x00018440, PHY_NUM1: 0x00018444, PHY_NUM2: 0x00018448
Offset Formula: 0x00018440+n*4: where n (0-2) represents PHY_NUM

Bit	Field	Type/InitVal	Description
1	Tx Bit Stuff En	RW 0x1	Indicates if the data on the DATA_OUT[15:0] lines needs to be bit-stuffed or not. Only used when opmode = 2'b11. 0 = Disable: Bit-stuffing is disabled. 1 = Enable: Bit-stuffing is enabled.
0	FS LS Serial Mode	RW 0x0	FS/LS Serial Mode Enable 0 = Disable: FS and LS packets are sent using the parallel interface. 1 = Enable: FS and LS packets are sent using the serial interface.

Table 1908: SDIO3 Configuration Register
Offset: 0x00018454

Bit	Field	Type/InitVal	Description
31:3	Reserved	RSVD 0x0	Reserved
2	Sample SD FeedBack Clock	RW 0x0	This bit should be set when working in SDR50 Mode, 100 MHz SD CLK. Should be reset in all other modes.
1	Reserved	RSVD 0x1	Reserved
0	SDIO3 Clock Inv	RW 0x1	SDIO3 Clock Invert. 0 = Not Inverted 1 = Inverted

Table 1909: QSGMII Control 1 Register
Offset: 0x00018494

Bit	Field	Type/InitVal	Description
31	QsgmiiRstn	RW 0x1	QSGMII reset, AL 0 = Reset mode 1 = Normal mode
30	QsgmiiActive	RW 0x0	QSGMII Active 0 = QSGMII Disconnected: SERDES is connected to the Gigabit Ethernet units. 1 = QSGMII Connected: QSGMII is connected to the Gigabit Ethernet units.
29	QsgmiiV3Active	RW 0x0	QSGMII Version3 Active 0 = QSGMII V3 Disconnected: SERDES is connected to the Gigabit Ethernet V3 units. 1 = QSGMII V3 Connected: QSGMII is connected to the Gigabit Ethernet V3 units.

Copyright © 2015 Marvell

Table 1909: QSGMII Control 1 Register (Continued)
Offset: 0x00018494

Bit	Field	Type/InitVal	Description
28	QsgmiiVersion	RW 0x0	QSGMII Version. Sets the version of QSGMII unit. 0 = Version1 1 = Version3
27:17	Reserved	RSVD 0x100	Reserved
16	Reserved	RW 0x0	Reserved
15	Reserved	RSVD 0x0	Reserved
14	UseCommaDetectSyncSM	RW 0x0	Use Comma Detect Sync SM Enables a second Sync SM in Comma Detect block - non standard mode (compatible with SATA mode) **NOTE:** This field may be changed only when <QsgmiiRstn> is HIGH. 0 = Disable 1 = Enable
13:0	Reserved	RSVD 0x0	Reserved

Table 1910: QSGMII Status Register
Offset: 0x00018498

Bit	Field	Type/InitVal	Description
31:8	Reserved	RSVD 0x0	Reserved
7:3	Reserved	RO 0x0	Reserved
2	Reserved	RSVD 0x0	Reserved
1	SyncOK	RO 0x0	Sync OK Rx synchronization SM has reached synchronization. 0 = Not synchronized 1 = Synchronized
0	CommaAligned	RO 0x0	Comma is detected. 0 = Not detected 1 = Detected

Copyright © 2015 Marvell

Table 1911: RTC SoC Interrupt Register
Offset: 0x000184A8

Bit	Field	Type/InitVal	Description
31:4	Reserved	RSVD 0x0	Reserved
3	RTC ALARM2 Interrupt Mask	RW 0x0	Mask bit for the corresponding interrupt. 0 = Mask: Interrupt is masked. 1 = Enable: Interrupt is enabled.
2	RTC ALARM1 Interrupt Mask	RW 0x0	Mask bit for the corresponding interrupt. 0 = Mask: Interrupt is masked. 1 = Enable: Interrupt is enabled.
1	RTC ALARM2 Interrupt	RW0C 0x0	RTC ALARM2 Interrupt.
0	RTC ALARM1 Interrupt	RW0C 0x0	RTC ALARM1 Interrupt.

Table 1912: SSCG Frequency Configuration Register
Offset: 0x000184AC

Bit	Field	Type/InitVal	Description
31:1	Reserved	RSVD 0x421815	Reserved
0	PLL Lock Detect	RO 0x0	PLL Lock Detect 0 = Locked 1 = Not Locked

Table 1913: SSCG SSC Mode Register
Offset: 0x000184B0

Bit	Field	Type/InitVal	Description
31	SSC Mode Select	RW 0x0	SSC Mode Select. 0 = Center Spread 1 = Down Spread
30:4	Reserved	RSVD 0xD05832	Reserved
3	SSC Reset	RW 0x1	SSC Reset 0 = Normal 1 = Reset
2	External PI Reset	RW 0x1	External Phase Interpolator (PI) Reset 0 = Normal 1 = Reset
1	PI Loop Control	RW 0x0	Phase Interpolator Loop Control 0 = Disable: PI is out of the PLL loop. 1 = Enable: PI is in the PLL loop.

Copyright © 2015 Marvell

Table 1913: SSCG SSC Mode Register (Continued)
Offset: 0x000184B0

Bit	Field	Type/InitVal	Description
0	SSC Clock Enable	RW 0x0	SSC Clock Enable. This selection enables the PI output clock for SSC digital logic. 0 = Disable 1 = Enable

Table 1914: LDO Control Register
Offset: 0x000184C0

Bit	Field	Type/InitVal	Description
31:26	Reserved	RSVD 0x0	Reserved
25:24	LdoB_SelTMux	RW 0x0	Analog test mux control signals Code TMUX_SEL[1:0] TMUX_OUT 2b00 Default Open Circuit (high impedance) 2b01 VREF (nominal 0.75 V) 2b10 VFB (nominal 0.75 V) 2b11 Internal 1.8 V supply divided by 2 (nominal 0.9 V)
23	LdoB2Rdy	RW 0x1	Logic Low enables the VOUT_SEL control word. Logic High ignores the VOUT_SEL control word and uses the default metal programmable setting of 2.5 V for the LDO output voltage level. 0 = Enable: VOUT_SEL is enabled. 1 = Disable: VOUT_SEL is disabled.
22:18	LdoBVoutSel	RW 0x18	Selects LDO output voltage level. Refer to LDO specification for coding definitions.
17	LdoBVdd3Rdy10	RO 0x0	Active High indicates that the LDO's 1V band gap voltage is ready.
16	LdoB1Rdy	RO 0x0	Active High indicates that the LDO output voltage level is valid.
15:10	Reserved	RSVD 0x0	Reserved
9:8	LdoA_SelTMux	RW 0x0	Analog test mux control signals Code TMUX_SEL[1:0] TMUX_OUT 2b00 Default Open Circuit (high impedance) 2b01 VREF (nominal 0.75 V) 2b10 VFB (nominal 0.75 V) 2b11 Internal 1.8 V supply divided by 2 (nominal 0.9 V)
7	LdoA2Rdy	RW 0x1	Logic Low enables the VOUT_SEL control word. Logic High ignores the VOUT_SEL control word and uses the default metal programmable setting of 1.8V for the LDO output voltage level. 0 = Enable: VOUT_SEL is enabled. 1 = Disable: VOUT_SEL is disabled.
6:2	LdoAVoutSel	RW 0xA	Selects LDO output voltage level. Refer to LDO specification for coding definitions.
1	LdoAVdd3Rdy10	RO 0x0	Active High indicates that the LDO's 1V band gap voltage is ready.

Copyright © 2015 Marvell

Table 1914: LDO Control Register (Continued)
Offset: 0x000184C0

Bit	Field	Type/InitVal	Description
0	LdoA1Rdy	RO 0x0	Active High indicates that the LDO output voltage level is valid.

Table 1915: IO Configuration 1 Register
Offset: 0x000184E4

Bit	Field	Type/InitVal	Description
IO CMOS Pads ZNR/ZPR configuration. **NOTE:** Reserved for Marvell use. Do not change the default value without approval from a Marvell Field Applications Engineer.			
31:28	SEG D MUST PADS ZPR	RW 0xb	IO Segment ZPR value.
27:24	SEG D MUST PADS ZNR	RW 0xb	IO Segment ZNR value.
23:20	SEG C MUST PADS ZPR	RW 0xb	IO Segment ZPR value.
19:16	SEG C MUST PADS ZNR	RW 0xb	IO Segment ZNR value.
15:12	SEG B MUST PADS ZPR	RW 0xb	IO Segment ZPR value.
11:8	SEG B MUST PADS ZNR	RW 0xb	IO Segment ZNR value.
7:4	SEG A MUST PADS ZPR	RW 0xb	IO Segment ZPR value.
3:0	SEG A MUST PADS ZNR	RW 0xb	IO Segment ZNR value.

Table 1916: IO Configuration 2 Register
Offset: 0x000184E8

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved
15:12	SEG E MUST PADS ZPR	RW 0xb	See MISC MUST PADS ZNR.
11:8	SEG E MUST PADS ZNR	RW 0xb	MISC_MUST - includes JTAG signals and SYSRSTn.
7:0	Reserved	RSVD 0xBB	Reserved

Table 1917: PCIe01 Reference Clock Buffer Control Register
Offset: 0x000184F0

Bit	Field	Type/InitVal	Description
1	PCIe1 Power UP	RW 0x1	PCIe Reference Clock Buffer Power Up Control 0 = Power Down 1 = Power Up
0	PCIe0 Power Up	RW 0x1	PCIe Refernce Clock Buffer Power UP Control 0 = Power Down 1 = Power Up

Table 1918: IO Configuration 4 Register
Offset: 0x000184F4

Bit	Field	Type/InitVal	Description
31:30	I2C PAD SR	RW SAR	Controls MPP[3:2] slew rate and output impedance, where 0 is the weakest and 3 is the strongest.
29:28	I2C Select	RW SAR	Enables I2C mode that affects the performance of the MPP[3:2] pads. **NOTE:** Values not shown should not be used. 0 = Non-I2C or I2C Standard Mode 1 = Fast Mode
27:0	Reserved	RSVD 0x210842	Reserved

A.29.3 Power Management and Memory Power Down

Table 1919: Power Management Clock Gating Control Register
Offset: 0x00018220

Bit	Field	Type/InitVal	Description
Core clock gating			
31	EIP150F Core Clock Gate	RW 0x1	EIP150F Clock Gate 0 = Clock Disable 1 = Clock Enable
30	Sata23 Core Clock Gate	RW 0x1	SATA Port2 and Port3 Core Clock Gate 0 = Clock Disable 1 = Clock Enable
29	PNC Clock Gate	RW 0x1	PNC Clock Gate 0 = Clock Disable 1 = Clock Enable
28	XOR1 Clock Gate	RW 0x1	XOR1 Clock Gate 0 = Clock Disable 1 = Clock Enable

Document Classification: Proprietary Information
Copyright © 2015 Marvell

Table 1919: Power Management Clock Gating Control Register (Continued)
Offset: 0x00018220

Bit	Field	Type/InitVal	Description
27:26	Reserved	RSVD 0x0	Reserved
25	TDM Clock Gate	RW 0x1	TDM Clock Gate 0 = Clock Disable 1 = Clock Enable
24	Reserved	RSVD 0x1	Reserved
23	Crypto0 Clock Gate	RW 0x1	Cryptographic0 Clock Gate 0 = Clock Disable 1 = Clock Enable
22	XOR0 Clock Gate	RW 0x1	XOR0 Clock Gate 0 = Clock Disable 1 = Clock Enable
21	Crypto1 Clock Gate	RW 0x1	Cryptographic1 Clock Gate 0 = Clock Disable 1 = Clock Enable
20:19	Reserved	RSVD 0x0	Reserved
18	USB2 Clock Gate	RW 0x1	USB2 Clock Gate 0 = Clock Disable 1 = Clock Enable
17	SDIO Clock Gate	RW 0x1	SDIO Clock Gate 0 = Clock Disable 1 = Clock Enable
16	Crypto1 Z Clock Gate	RW 0x1	Cryptographic1 Z Clock Gate 0 = Clock Disable 1 = Clock Enable
15	Sata01 Core Clock Gate	RW 0x1	SATA Port0 and Port1 Core Clock Gate 0 = Clock Disable 1 = Clock Enable
14	Crypto0 Z Clock Gate	RW 0x1	Cryptographic0 Z Clock Gate 0 = Clock Disable 1 = Clock Enable
13	BM Clock Gate	RW 0x1	Buffer Management Clock Gate 0 = Clock Disable 1 = Clock Enable
12	Reserved	RW 0x1	Reserved
11	USB3D Clock Gate	RW 0x1	USB3 Device Clock Gate 0 = Clock Disable 1 = Clock Enable
10	USB3H1 Clock Gate	RW 0x1	USB3 Host 1 Clock Gate 0 = Clock Disable 1 = Clock Enable

Copyright © 2015 Marvell

Table 1919: Power Management Clock Gating Control Register (Continued)
Offset: 0x00018220

Bit	Field	Type/InitVal	Description
9	USB3H0 Clock Gate	RW 0x1	USB3 Host 0 Clock Gate 0 = Clock Disable 1 = Clock Enable
8	PCIe0 Clock Gate	RW 0x1	PCIe0 Clock Gate 0 = Clock Disable 1 = Clock Enable
7	PCIe3 Clock Gate	RW 0x1	PCIe3 Clock Gate 0 = Clock Disable 1 = Clock Enable
6	PCIe2 Clock Gate	RW 0x1	PCIe2 Clock Gate 0 = Clock Disable 1 = Clock Enable
5	PCIe1 Clock Gate	RW 0x1	PCIe1 Clock Gate 0 = Clock Disable 1 = Clock Enable
4	GbE0 Clock Gate	RW 0x1	GbE0 Clock Gate 0 = Clock Disable 1 = Clock Enable
3	GbE1 Clock Gate	RW 0x1	GbE1 Clock Gate 0 = Clock Disable 1 = Clock Enable
2	GbE2 Clock Gate	RW 0x1	GbE2 Clock Gate 0 = Clock Disable 1 = Clock Enable
1	Reserved	RSVD 0x1	Reserved
0	Audio Clock Gate	RW 0x1	Audio Clock Gate 0 = Clock Disable 1 = Clock Enable

A.29.4 BootROM

Table 1920: BootROM Routine and Error Code Register
Offset: 0x000182D0

Bit	Field	Type/InitVal	Description
31:28	Reserved	RW 0xf	Reserved Write only 0xF.
27:24	Reserved	RW 0xf	Reserved Write only 0xF.
23:16	Retry_Count	RW 0xFF	The retry count is incremented in the Error Handler routine.

Copyright © 2015 Marvell

Table 1920: BootROM Routine and Error Code Register (Continued)
Offset: 0x000182D0

Bit	Field	Type/InitVal	Description
15	Reserved	RW 0x0	Reserved Write only 0x0.
14:13	Reserved	RW 0x0	Reserved Write only 0x0.
12	Reserved	RW 0x0	Reserved Write only 0x0.
11:8	Error Location	RW 0x0	Error Location Values not listed are reserved. 0 = Initialization 4 = SATA: Boot from SATA interface. 7 = SPI: Boot from SPI interface. 9 = NAND Flash: Boot from NAND Flash interface. 11 = Exception: Exception handler 12 = Main: Main execution loop 14 = PCI Express: Boot from PCI Express interface.

Copyright © 2015 Marvell

Table 1920: BootROM Routine and Error Code Register (Continued)
Offset: 0x000182D0

Bit	Field	Type/InitVal	Description
7:0	Error Code	RW 0x0	Error Code: Values not listed are reserved. 0x0 = No Error 0x11 = Invalid header ID 0x12 = Invalid header checksum 0x13 = Invalid image checksum 0x15 = Invalid extended header checksum 0x17 = SATA device is busy. 0x18 = SATA device link error 0x19 = SATA DMA transfer error 0x1A = SATA PIO transfer error 0x1C = Invalid header size 0x1D = Unknown boot device 0x1E = Unsupported reset strap option 0x21 = Image size not aligned to 32 bits 0x22 = Source address not aligned to 32 bits 0x23 = Destination address not aligned to 32 bits 0x30 = Failed to read from NAND flash 0x31 = Invalid bootROM checksum 0x32 = NAND flash image not aligned to 512 bytes 0x33 = NAND flash ECC error 0x34 = NAND flash timeout 0x35 = NAND flash excessive bad blocks 0x36 = NAND flash bad block encountered 0x37 = NAND flash unsupported ECC type 0x51 = Invalid security header 0x52 = Invalid RSA pubic key 0x53 = Invalid RSA public key FMT 0x54 = Invalid RSA header signature 0x55 = Invalid RSA image signature 0x56 = Invalid box ID 0x57 = Invalid flash ID 0x61 = Invalid binary header 0x62 = Invalid image length 0x63 = Invalid header length 0x1B = I2C timeout (Reserved) 0x40 = I2C read error (Reserved) 0x41 = I2C write error (Reserved) 0x42 = I2C start error (Reserved) 0x43 = I2C stop error (Reserved) 0x44 = I2C no header found (Reserved) 0x58 = RSA library error (Reserved)

A.29.5 Interrupt

Table 1921: Misc Error Interrupt Cause Register
Offset: 0x000182A8

Bit	Field	Type/InitVal	Description
31:16	Reserved	RSVD 0x0	Reserved

Copyright © 2015 Marvell

Table 1921: Misc Error Interrupt Cause Register (Continued)
Offset: 0x000182A8

Bit	Field	Type/InitVal	Description
15	USB3 Device TX Parity Error	RW0C 0x0	USB3 Device TX RAM Parity Error Interrupt
14	USB3 Device RX Parity Error	RW0C 0x0	USB3 Device RX RAM Parity Error Interrupt
13	Reserved	RSVD 0x0	Reserved
12	CPU1 SCU Parity Fail Error	RW0C 0x0	CPU1 SCU Parity Fail Error Interrupt
11	CPU0 SCU Parity Fail Error	RW0C 0x0	CPU0 SCU Parity Fail Error Interrupt
10	SCUEV Abort Error	RW0C 0x0	Indicates an external abort has occurred during a coherency writeback.
9:8	Reserved	RSVD 0x0	Reserved
7	SDIO3 Parity Error	RW0C 0x0	SDIO3 Parity Error Interrupt
6	Audio Parity Fail Error	RW0C 0x0	Audio IP RAM Parity Error Interrupt
5	SATA1 Mbus Parity Error	RW0C 0x0	SATA Port 1 Mbus Parity Error Interrupt
4	SATA0 Mbus Parity Error	RW0C 0x0	SATA Port 0 Mbus Parity Error Interrupt
3	SATA1 CH1 Parity Error	RW0C 0x0	SATA Port 1 Channel 1 Parity Error Interrupt
2	SATA1 CH0 Parity Error	RW0C 0x0	SATA Port 1 Channel 0 Parity Error Interrupt
1	SATA0 CH1 Parity Error	RW0C 0x0	SATA Port 0 Channel 1 Parity Error Interrupt
0	SATA0 CH0 Parity Error	RW0C 0x0	SATA Port 0 Channel 0 Parity Error Interrupt

Table 1922: Misc Error Interrupt Mask Register
Offset: 0x000182AC

Bit	Field	Type/InitVal	Description
31:12	Reserved	RSVD 0x0	Reserved
11:0	Misc Error Interrupt Mask	RW 0x0	Mask bit per bit in <Misc Error Interrupt Cause> Register

Copyright © 2015 Marvell

A.29.6 Design for Testablity (DFT)

Table 1923: Sample at Reset (SAR) Register
Offset: 0x00018600

Bit	Field	Type/InitVal	Description
31	Rst2Core PCIe3 RefClk Out Enable	RO 0x0	PCIe3 RefClk Out Enable **NOTE:** Internally pulled to 0x1. 0 = Disable: PCIe clock output disabled: The device uses an external source for PCI Express clock. Pins PCIe3_CLK_N and PCIe3_CLK_P are inputs. 1 = Enable: PCIe clock output enable: The device uses an internally generated clock for PCI Express clock. Pins PCIe3_CLK_N and PCIe3_CLK_P are outputs, driving out the PCI Express differential clock.
30	Rst2Core PCIe2 RefClk Out Enable	RO 0x0	PCIe2 RefClk Out Enable **NOTE:** Internally pulled to 0x1. 0 = Disable: PCIe clock output disabled: The device uses an external source for PCI Express clock. Pins PCIe2_CLK_N and PCIe2_CLK_P are inputs. 1 = Enable: PCIe clock output enable: The device uses an internally generated clock for PCI Express clock. Pins PCIe2_CLK_N and PCIe2_CLK_P are outputs, driving out the PCI Express differential clock.
29:27	Rst2Core Device ID	RO 0x0	Device ID reset strap.
26:22	Reserved	RSVD 0x0	Reserved
21	Rst2PLL SSCG Bypass	RO 0x0	SSCG Bypass Control 0 = Normal 1 = Bypassed
20:19	Reserved	RSVD 0x0	Reserved
18	Rst2Pr Except Hand Thumb Enable	RO 0x0	Exception Handler in ARM/Thumb mode 0 = ARM mode 1 = Thumb mode
17	Rst2Pr FIQ SW Mask Disable	RO 0x0	Disable Fast Interrupt Software Masking 0 = Masking: SW masking of FIQ is enabled. 1 = No masking: SW cannot mask FIQ.
16	Rst2Pr Except Hand Endian Mode Sel	RO 0x0	Exception Handler Endianness This bit of the CPSR controls the load and store endianness for data processing by the exception handler. 0 = Little Endian: Exception handler data is in Little Endian mode. 1 = Big Endian: Exception handler data is in Big Endian mode.
15	Rst2 Core Clock Select	RO 0x0	Determines the frequency of the core clock. 0 = 250 MHz 1 = 200 MHz

Copyright © 2015 Marvell
Document Classification: Proprietary Information

Table 1923: Sample at Reset (SAR) Register (Continued)
Offset: 0x00018600

Bit	Field	Type/InitVal	Description
14:10	Rst2 CPU DDR Clock Ratio Select	RO 0x0	Selects the ratios between PCLK, NBCLK, HCLK, and DRAM clocks. Additional test options are available for testing purposes. For full details about the various options, refer to the Clocking section in the Hardware Specifications. **NOTE:** Internally pulled to a value of 0xC.
9:4	Rst2R Boot Mode Select	RO 0x0	See Boot Option description in the device Hardware Specifications. **NOTE:** Internally pulled to a value of 0x36.
3	Rst2Core PCIe1 RefClk Out Enable	RO 0x0	PCIe1 RefClk Out Enable **NOTE:** Internally pulled to 0x1. 0 = Disable: PCIe clock output disabled: The device uses an external source for PCI Express clock. Pins PCIe1_CLK_N and PCIe1_CLK_P are inputs. 1 = Enable: PCIe clock output enable: The device uses an internally generated clock for PCI Express clock. Pins PCIe1_CLK_N and PCIe1_CLK_P are outputs, driving out the PCI Express differential clock.
2	Rst2Core PCIe0 RefClk Out Enable	RO 0x0	PCIe0 RefClk Out Enable **NOTE:** Internally pulled to 0x1. 0 = Disable: PCIe clock output disabled: The device uses an external source for PCI Express clock. Pins PCIe0_CLK_N and PCIe0_CLK_P are inputs. 1 = Enable: PCIe clock output enable: The device uses an internally generated clock for PCI Express clock. Pins PCIe0_CLK_N and PCIe0_CLK_P are outputs, driving out the PCI Express differential clock.
1	Rst2 RefClkOut Enable	RO 0x0	Configure reference output clock frequency and mode. Reset to RefClkOut Enable **NOTE:** Once Enabled, MPP[46:45] wakes up as REF_CLK_OUT[1:0]. **NOTE:** Internally pulled to 0x1. 0 = Disable 1 = Enable: Ref_Clk_Out 0 running at 25 MHz, Ref_Clk_Out 1 running at 25 MHz
0	Rst2R I2C Init Enable	RO 0x0	I2C0 Serial ROM Enable **NOTE:** When I2C0 Serial ROM is enabled, MPP[3:2] wakes up as I2C Clock and Data. **NOTE:** Internally pulled to 0x0. 0 = Disable 1 = Enable

Copyright © 2015 Marvell

Table 1924: PUP Enable Register
Offset: 0x0001864C

Bit	Field	Type/InitVal	Description
31:6	Reserved	RSVD 0x2007FEF	Reserved
5	Spi PUP Enable	RW SAR	SPI PUP Enable. Must be set to 0x1. Enabling this bit must be done after setting relevant MPP selectors.
4	Reserved	RSVD 0x1	Reserved
3	Device PUP Enable	RW SAR	Device PUP Enable. Must be set to 0x1. Enabling this bit must be done after setting relevant MPP selectors.
2:0	Reserved	RSVD 0x3	Reserved

Table 1925: On-Die Pattern Generator (ODPG) Enable Register
Offset: 0x000186D4

Bit	Field	Type/InitVal	Description
31:10	Reserved	RSVD 0x0	Reserved
9	DDR ODPG Done	RW0C 0x0	DDR ODPG Done Indication.
8	OdpgDis	RW 0x0	DRAM ODPG Disable. A write of 1 terminates ODPG operation. This field is cleared by HW when the ODPG actually terminates. A write of 0 has no affect.
7:1	Reserved	RSVD 0x0	Reserved
0	OdpgEn	RW 0x0	DRAM ODPG Enable. A write of 1 enables ODPG. This field is cleared by HW when the ODPG sequence completes (SW may use read polling to detect completion). A write of 0 has no affect. **NOTE:** Do not access the DRAM as long as the ODPG is running.

Copyright © 2015 Marvell

B Acronyms and Abbreviations

Numeric	3GPP	3rd Generation Partnership Project
A	ACK	Acknowledge
	AES	Advanced Encryption Standard
	AES128/128	128 data bits AES with 128-bit key width
	AES128/192	128 data bits AES with 192-bit key width
	AES128/256	128 data bits AES with 256-bit key width
	AGCH	Access Grant Channel
	AI	Acquisition Indication
	AICH	Acquisition Indicator Channel
	AMP	Asymmetric Multiprocessor
	AMR	Adaptive Multi-Rate (vocoder)
	API	Application Programming Interface
	ARFCN	Absolute Radio Frequency Channel Number
	ASC	Access Service Class
	ASM	Any Source Multicast
B	BM	Buffer Management
	BAR	Base Address Register
	BCCH	Broadcast Control Channel
	BCH	Broadcast Channel
	BER	Burst Error Rate
	BIP	Bit Interleave Parity
	BLER	Block Error Ratio
	BM	Buffer Management
	BPPE	External Buffer Pointers Pool
	BPPI	Internal Buffer Pointers Pool
	BSIC	Base Transceiver Station Identity Code
	BTAC	Branch Target Address Cache
	BW	Bandwidth

Copyright © 2015 Marvell

C	CB	Cell Broadcast
	CBC	Cipher Block Chain
	CBCH	Cell Broadcast Channel
	CBS	Cell Broadcast Service
	CCCH	Common Control Channels
	CCPCH	Common Control Physical Channel
	CDMA	Code Division Multiple Access
	CESA	Cryptographic Engine Security Accelerator
	CFB	Cipher Feedback
	CFN	Connection Frame Number
	CIR	Committed Information Rate
	CnM	Classification and Marking
	CRC	Cyclic Redundancy Check
	CS	Circuit Switched
	CTCH	Common Traffic Channel
D	DB	Data Base
	DBR	Dynamic Bandwidth Report
	DBRu	Dynamic Bandwidth Report upstream
	DBX	Direct Bytecode eXecution
	DCCH	Dedicated Control Channel
	DCH	Dedicated Channel
	DES	Data Encryption Standard
	3DES	Triple Data Encryption Standard
	DL	Down Link
	DP	Drop Precedence
	DPCCH	Dedicated Physical Control Channel
	DPCH	Dedicated Physical Channel
	DRX	Discontinuous Receiving
	DS	Downstream
	DSCP	Differentiated Services Code Point
	DTX	Discontinuous Transmission
E	EB	Eviction Buffers
	EBS	Excessive Burst Size
	ECB	Electronic Code Book
	ECC	Error-Correcting Code

Copyright © 2015 Marvell

	Ec/N_o	Ratio of energy per modulation bit to the noise spectral density
	ED-AQM	Early Drop Active Queue Management
	EDE	Encryption Decryption Encryption
	EDMA	Enhanced-DMA
	EDRP_CFG	Early Drop Configuration
	EEE	Encryption Encryption Encryption (when referring to the CESA)
		Energy Efficient Ethernet (when referring to Ethernet)
	EFC	Ethernet FCS Checker
	eFuse	Electronic Fuse
	EGPRS	EDGE General Packet Radio Service
	EJP	Enhanced Jitter Pacing
	ELF	Executable and Linkable Format
	EMR	Enhanced Measurement Reporting
	EP	Endpoint
F	FACCH	Fast Associated Control Channel
	FACH	Forward Access Channel
	FB	Frequency Correction Burst
	FCCH	Frequency Correction Channel
	FCS	Frame Check Sequence
	FEC	Forward Error Correction
	FFS	For Further Study
	FIFO	First In First Out
	FIQ	Fast Interrupt Request
	FPDMA	First party DMA
	FPGA	Field Programmable Gate Array
	FPU	Floating Point Unit
G	GbE	Gigabit Ethernet
	GHB	Global History Buffer
	GIC	Generic Interrupt Controller
	GPRS	General Packet Radio Service
	GSM	Global System for Mobile Communication
H	HHO	Hard Hand Over
	HPLMN	Home PLMN
	HSL	High-Speed Logger

Copyright © 2015 Marvell

	HUM	Hit Under Miss
	HWF	Hardware Forwarding
I	I^2C	Inter-Integrated Circuit
	I^2S	Integrated Interchip Sound
	IBC	Internal Buffer Control
	ICE	In-Circuit Emulator
	IDB	Internal Data Buffer or Internal-packet Data Buffer
	IGMP	Internet Group Management Protocol
	IMS	Interface Management Subsystem
	IMSI	International Mobile Subscriber Identity
	I/O	Input/Output
	IOCC	Input/Output Cache Coherency
	IPG	inter-packet gap
	IRQ	Interrupt Request
	IV	Initial Vector/Initial Value
L	L1	Layer 1 or Level 1 (Level 1 when referring to cache)
	L2	Layer 2 or Level 2 (Level 2 when referring to cache)
	LFB	Line Fill Buffers
	LLC	Logical Link Control
	LPI	Low Power Idle
	LRB	Line Read Buffers
	LUT	Look Up Table
M	MAC	Medium Access Control
	MC	Multicast
	MD5	Message Digest 5
	MMU	Memory Management Unit
	MPE	Media Processing Engine
	MPP	Multipurpose Pins
	MS	Mobile Station
	MSC	Message Sequence Chart
	MSI	Message Signaled Interrupt
N	N/A	Not Applicable
	NACK	Negative ACK

Copyright © 2015 Marvell

	NAPT	Network Address and Protocol Translation
	NAS	Network Attached Storage
	NCQ	Native Command Queuing
	NFC	NAND Flash Controller
	NFP	Network Fast Processing
	Node B	A logical node responsible for radio transmission / reception in one or more cells to/from the User Equipment.
	NVM	Non-Volatile Memory
O	OFB	Output Feedback
	OOB	Out of Band
P	P2P	Point-to-Point
	P2PE	Point-to-Point Emulation
	PBCCH	Packet Broadcast Control Channel (GPRS)
	PCCCH	Packet Common Control Channel (GPRS)
	P-CCPCH	Primary Common Control Channel
	PCH	Paging Channel
	PCIe	PCI Express
	PCS	Physical Coding Sublayer
	PDB	Plend Decode Block
	PDCH	Packet Data Channel (GPRS)
	PDSCH	Physical Downlink Shared Channel
	PDTCH	Packet Data Traffic Channel (GPRS)
	PDU	Protocol Data Unit
	PI	Paging Indication
	P. Idle	Packet Idle
	PICH	Paging Indication Channel
	PIPE	PHY Interface for PCI Express
	PIPT	Physical Indexed Physically Tagged
	PLMN	Public Land Mobile Network
	PLP	Physical Layer Processor
	PM	Port Multiplier
	PMU	Performance Monitor Unit
	PoE	Power Over Ethernet
	PPI	private peripheral interrupts
	PPPoE	Point-to-Point Protocol over Ethernet

Copyright © 2015 Marvell

	PRACH	Packet Random Access Channel
	PS	Port Selector (when referring to SATA)
		Protocol Stack (Layers 2/3)
	PSA	Physical Synchronization and Alignment
	PSI	Packet System Information
	PTCCH	Packet Timing Advance Control Channel (GPRS)
	PTM	Packet Transfer Mode
	PTP	Precise Time Protocol
	PUP	Pack/Unpack
Q	Qos	Quality of Service
R	RA	Read allocate
	RACH	Random Access Channel
	RAID	Redundant Array of Independent Disks
	RAT	Radio Access Technology (UMTS)
	RC	Root Complex
	RDI	Rx DMA Interface
	RF	Radio Frequency
	RL	Radio Link
	RLC	Radio Link Control (GPRS/UMTS)
	RR	Radio Resource sublayer
	RRC	Radio Resource Control
	RS	Reed-Solomon
	RSCP	Received Signal Code Power
	RSS	Receive Side Scaling
	RSSI	Radio Signal Strength Indicator
	RTC	Real Time Clock
	Rx	Receiver
S	SACCH	Slow Associated Control Channel
	SAR	Sample at Reset
	SATA	Serial Advanced Technology Attachment (ATA)
	SB	Synchronization Burst
	SC	Scrambling Code
	SCB	Single Copy Broadcast
	SCC	Serving Cell Change

Copyright © 2015 Marvell

	S-CCPCH	Secondary Common Control Channel
	SCU	Snoop Control Unit
	SDCCH	Stand-alone Dedicated Control Channel
	SDU	Service data unit
	SECDEC	Single Error Correction Double Error Detection
	SFN	System Frame Number
	SFU	Single Family Unit
	SGI	software generated interrupts
	SHA-1	Secure Hash Algorithm 1
	SHA-2	Secure Hash Algorithm 2
	SHA-256	SHA-2 with 256 digest bit size
	SHO	Soft Handover
	SI	System Information
	SIMD	Single Instruction Multiple Data
	SMP	Symmetric Multiprocessing
	SPI	Serial Peripheral Interface
	SP / WRR	Static Priority / Weighted Round-Robin
	S/PDIF	Sony/Philips Digital Interconnect Format
	SSM	Source Specific Multicast
	STB	Store Buffers
	STTD	Space Time Transmit Diversity
	SWF	Software Forwarding
T	TA	Timing Advance
	TBD	To Be Defined
	TBF	Temporary Block Flow (GPRS)
	TC	Transmission Convergence
	TCH	Traffic Channel
	TCP	Transmission Control Protocol
	TDM	Time-Division Multiplexing
	TF	Transport Format
	TFC	Transport Format Combination
	TFCS	Transport Format Combination Set
	TFI	Temporary Frame Identity (GPRS)/ Indicator (UMTS)
	TFS	Transport Format Set
	TGCFN	Transmission Gap Connection Frame Number
	TLB	Transaction Look-ahead Buffers

Copyright © 2015 Marvell

	TMSI	Temporary Mobile Subscriber Identity
	TrCH	Transport Channel
	TTI	Transmit Timing Interval
	TS	Technical Specification
	Tx	Transmitter
U	UARFCN	UTRA Absolute Radio Frequency Channel Number
	UART	Universal Asynchronous Receiver/Transmitter
	UC	Unicast
	UDP	User Datagram Protocol
	UE	User Equipment
	UL	Up Link
	UMTS	Universal Mobile Telecommunications System
	US	Upstream
	USF	Uplink State Flag (GPRS)
	UTRAN	Universal Terrestrial Radio Access Network
V	VC	Virtual Channel
	VIPT	Virtual Indexed Physically Tagged
W	W0…W15	Designates the 16 words in a CESA authentication input data block; W0 is the first word and W15 is the last word.
	WA	Write allocate
	WB	Write-Back
	WCDMA	Wideband Code Division Multiple Access
	WOL	Wake On LAN
	WRR	Weighted Round-Robin
	WT	Write-Through
X	xHCI	Extensible Host Controller interface

Copyright © 2015 Marvell

Copyright © 2015 Marvell



Marvell Technology Group
http://www.marvell.com

Marvell. Moving Forward Faster