

PM8001

Tachyon SPC 8x6G

8-Port 6 Gbit/s SAS/SATA Controller

Programmers Manual

Released

Issue No. 11: November 2010

Legal Information

Copyright

Copyright 2010 PMC-Sierra, Inc. All rights reserved.

The information in this document is proprietary and confidential to PMC-Sierra, Inc., and for its customers' internal use. In any event, no part of this document may be reproduced or redistributed in any form without the express written consent of PMC-Sierra, Inc.

PMC-2080222 (R11)

Disclaimer

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

Trademarks

For a complete list of PMC-Sierra's trademarks and registered trademarks, visit:
<http://www.pmc-sierra.com/legal/>.

Patents

The technology discussed in this document is protected by the following patent grants: U.S. Patent No. 7, 200, 170 B1 and 6, 293, 055 B2. Other relevant patent grants may also exist.

Contacting PMC-Sierra

PMC-Sierra
8555 Baxter Place
Burnaby, BC
Canada V5A 4V7

Tel: +1 (604) 415-6000
Fax: +1 (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Technical Support: apps@pmc-sierra.com
Web Site: <http://www.pmc-sierra.com>

Revision History

Issue No.	Issue Date	Details of Change
11	November 2010	<p>Section 3: Added the following new sections:</p> <ul style="list-style-type: none"> • 3.24.5 RETURN_ERROR_ON_OUT_OF_IOST_RESOURCE • 3.24.6 ENABLE_OUTBOUND_PROCESSING_FAIRNESS <p>Section 11:</p> <ul style="list-style-type: none"> • Section 11.4.2 Soft Reset Recovery (Normal Mode), added option to skip step 1 of the Soft Reset Sequence (Normal Mode).
10	June 2010	<p>Preface:</p> <ul style="list-style-type: none"> • Updated some reference document revisions. <p>Section 3:</p> <ul style="list-style-type: none"> • 3.2 Device Handle and DEVICE_ID, Added description of validity check of DEVICE_ID. • 3.24.3 IOP_EVENT_PHY_LOCKUP_DATAOUT, New section. • 3.24.4 IOP_EVENT_PHY_DOC_ABORT_TMO, New section. <p>Section 5:</p> <ul style="list-style-type: none"> • 5.2.1 MPI Main Configuration Table Fields, Updated description of IO Abort Delay. • 5.2.4 MPI Outbound Queue Configuration Table Fields, Changed description of Outbound Queue n Interrupt Mode. <p>Section 7:</p> <ul style="list-style-type: none"> • 7.2 PHY_START Command, Added SSC Disable to PHY_START Command Format and Field Description table. <p>Section 8:</p> <ul style="list-style-type: none"> • 8.2 SAS_HW_EVENT Notification, Added “Optional” to IOP_EVENT_IT_NEXUS_LOSS. Added the following new errors: IOP_EVENT_PHY_LOCKUP_DATAOUT and IOP_EVENT_PHY_DOC_ABORT_TMO. • 8.3 SSP_COMPLETION Response, Adding this revision history item to document that the description of 0x00000012: IO_OPEN_CNX_ERR_BREAK was updated in the last issue of this manual. <p>Section 9:</p> <ul style="list-style-type: none"> • 9.1 Programming Interface, Added potential PCIe unit reset when writing to a reserved register. • 9.2.48 Uncorrectable Error Status Register, Added note to Completion Timeout Status bit that completion timeout value is 8.5-10ms. <p>Section 10:</p> <ul style="list-style-type: none"> • 10.2.8 Scratchpad 3 Register, Updated description for HDA mode. Added new table “Scratchpad 3 Register when IOP_STATE is in

Issue No.	Issue Date	Details of Change
		<p>Ready State".</p> <ul style="list-style-type: none"> • 10.3.7 SPC Reset Register, In the SW_DEVICE_RSTB bit description, added "except in the case of a fatal error" to the customer warning. • 10.4.2 Lane N TRS Control Register, Updated definitions of TRS0_PREEN_50G and TRS0_PREEN_25G. • 10.5.4 Transmitter Per Port Configuration 1 SAS_SATA G1 Register, Updated description of T_MODE_13_12_SATA_G1 and T_MODE_13_12_SAS_G1. Changed default value. Changed reserved bits [25:24] to T_CTRL_5_4_SATA_G1. Changed reserved bits [9:8] to T_CTRL_5_4_SAS_G1. Changed reserved bit 13 to TRS_SSCEN_SAS_G1. • 10.5.5 Transmitter Per Port Configuration 1 SAS_SATA G2 Register, Updated description of T_MODE_13_12_SATA_G2. Changed default value. Changed reserved bits [25:4] to T_CTRL_5_4_SATA_G2. Updated description of bits [15:14] T_MODE_13_12_SAS_G2. Changed default value. Changed reserved bits [9:8] T_CTRL_5_4_SAS_G2. Changed reserved bit 13 to TRS_SSCEN_SAS_G2 • 10.5.6 Transmitter Per Port Configuration 1 SAS_SATA G3 Register, Updated description of T_MODE_13_12_SAS_G3. Changed default value. Changed reserved bits [25:24] to T_CTRL_5_4_SATA_G3. Updated description of T_MODE_13_12_SAS_G3. Changed default value. Changed reserved bits [9:8] to T_CTRL_5_4_SAS_G3. Changed reserved bit 13 to TRS_SSCEN_SAS_G3. • 10.5.7 Receiver Per Port Configuration 1 SAS_SATA G1G2 Register, Combined reserved bit 25 with 26 [25:26] R_RXMODE_14_13_SATA_G1G2. Combined reserved bit 10 with 9 [10:9] R_RXMODE_14_13_SAS_G1G2. • 10.5.8 Receiver Per Port Configuration 1 SAS_SATA G3 Register, Combined bits [26:25] into R_RXMODE_14_13_SATA_G3. Combined bits [10:9] into R_RXMODE_14_13_SAS_G3. • 10.5.10 Transmitter Configuration 1 Register, Changed T_MODE_1_0 Edge Rates. • 10.6.22 Open Retry Interval Register Updated description of OPEN_RETRY_INTERVAL_REG. <p>Section 11:</p> <ul style="list-style-type: none"> • 11.4.2 Soft Reset Recovery (Normal Mode), Updated step 1 of the Soft Reset Sequence (Normal Mode). • 11.4.3 Soft Reset Recovery (HDA Mode), Updated Step 1 of the soft reset sequence for the host.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • 11.6.1.1 SAS PHY Not Ready, Updated section to delete OOB handshake. • 11.6.2.6 BREAK Received During Open Request, Added cross-reference to SSP_Completion Response to instructions on initiator mode.
9	December 2009	<p>Section 2:</p> <ul style="list-style-type: none"> • 2.3 Inbound Queues (IQs) and 2.4 Outbound Queues (OQs) changed the maximum number of queues from 32 to 64. <p>Section 3:</p> <ul style="list-style-type: none"> • Figure 15, Corrected SSP Target Write Operations Flow Diagram • Section 3.24, Added section on customizing firmware behavior. • Section 3.25, Added section on SGPIO operation. <p>Section 5:</p> <ul style="list-style-type: none"> • Section 5.2.1, Table 38 Updated description of IRAD bit in MPI Configuration Table. Changed 0x13 to Customization Settings. • Section 5.2.4, Table 41 Corrected offset calculations for the Outbound Queue Depth through Dynamic Interrupt Coalescing. Updated the description of Outbound Queue 0 Interrupt Coalescing Count. <p>Section 7:</p> <ul style="list-style-type: none"> • Section 7.2, Updated description of PHY_START command for the handling of the OBID. • Section 7.5, Table 54 Updated Task Management Function description. Added description of when SSP_INI_TM_START Command can abort an SSP I/O. • Section 7.10 Updated description to include new NOQ bit. Table 64 and Table 65 Added NOQ bit. • Section 7.17 Updated description to include new NOQ bit. Table 80 and Table 81 Added NOQ bit. • Section 7.26, Table 103 Updated description of PORT_OP 0x03 and PORT_OP 0x05. • Section 7.32, Table 115 Changed MAX_PORTS field length to 1 byte. Updated OPEN REJECT (RETRY) Command and Data Phase field descriptions. • Added sections 7.33 SGPIO Register Command and 7.34 PCIE_DIAG Execute Command. <p>Section 8:</p> <ul style="list-style-type: none"> • Section 8.2, Table 125 Added new event IOP_EVENT_IT_NEXUS_LOSS • Section 8.3, Table 127 Added AGRS field (Auto Good Response Status), IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT and IO_XFER_ERROR_UNEXPECTED_PHASE. Updated IO_ABORTED description.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Section 8.4, Table 129 Added status value for IO_INVALID_LENGTH. Added IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT. Updated IO_ABORTED description. • Section 8.9, Table 139 Updated IO_ABORTED description. Added IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT. • Section 8.11, Table 143 Changed Event Type description of IO_XFER_ERROR_UNEXPECTED_PHASE. • Section 8.35, New section for SGPIO Response. • Section 8.36, New section for PCIE_DIAG Execute Response. <p>Section 9:</p> <ul style="list-style-type: none"> • Section 9.2.3, Table Table 196, Updated reset value for the ROM Base Address. <p>Section 10:</p> <ul style="list-style-type: none"> • Section 10.3.13, Corrected IB_AXI_ADDR_2 [36:32] reset value. • Section 10.2.6 and 10.2.7 – Added definition of “soft reset”. • Section 10.5.4, 10.5.5 and 10.5.6 Changed the reserved bits 31:30 and 15:14 to T_MODE_13_12_(and the appropriate SAS or SATA and register suffix). • Section 10.6.22, Table 302 Open Retry Interval Bits-updated description. <p>Section 11:</p> <ul style="list-style-type: none"> • Section 11.2.15, . Added code to sample register dump. Added HSST Code and changes to PCIE App. • Added Section 11.4.1 Chip Status Determination Before Soft-Reset. • Section 11.4.3, updated step 1 of Soft Reset Sequence. • Section 11.5.2.3, Changed Wide Port Successful “Link/Hard reset” to “Link Reset”. (113280) • Section 11.5.2.4, Changed Wide Port Partial Successful “Link/Hard reset” to “Link Reset”. • Section 11.5.2.5, Changed Wide Port Unsuccessful “Link/Hard reset” to “Link Reset”. • Section 11.5.2.6, New section on wide port successful hard reset. • Section 11.6.2.6 Updated section BREAK Received During Open Request • Section 11.6.2.10 ACK/NAK Timeouts, Updated description. • Section 11.6.5.7, Changed response message.
8	July 2009	<p>Section 3:</p> <ul style="list-style-type: none"> • Section 3.17.1, Table 15, Event Log Header Format, updated the DWord offset 0x00 for processor signature and 0x03 Event Log Buffer Size description. <p>Section 4:</p> <ul style="list-style-type: none"> • Section 4.3.1, Added Table 28 General Configuration

Issue No.	Issue Date	Details of Change
		<p>Section 5:</p> <ul style="list-style-type: none"> Section 5.1, Added Reserved bit to Table 38 MPI Configuration Table – Main Part Section 5.2.1, Table 38 MPI Configuration Table, deleted 0x1C [0] and 0x1C [6:1]; added 0x1C [1] and 0x1C [6:2]. Updated description of IOABTDLY. Added Reserved bits. Section 5.2.2, Added new field in HMI_ERR [6:0]. Corrected duplicate 'Recoverable Error Information 6' field. Section 5.2.6.1 Added new step after step 2. Various changes in register formats per customer. <p>Section 7:</p> <ul style="list-style-type: none"> Section 7.3 Corrected event code. Added field to ERR_CNT_RESET in Table 94 SAS Diagnostic Command Type and Command Description Valid Combination. Section 7.8, Table 60 Updated Override Device State field description. Section 7.10 and 7.17 Added information on delay time. Table 77 SATA_HOST_IO_START Command Format and Table 78 SATA_HOST_IO_START Command Fields: Added RETFIS (Return FIS on Good Completion). Section 7.23, Table 94 SAS Diagnostic Command Type and Command Description Valid Combination, Added field Section 7.32, Table 115, Changed Open Reject entries default to 256 <p>Section 8:</p> <ul style="list-style-type: none"> Section 8.3 Updated description of 0x0000000E: IO_XFER_ERROR_BREAK and 0x00000012: IO_OPEN_CNX_ERR_BREAK. Added 0x0000003F IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY. Section 8.9 Removed 0x0000000E: IO_XFER_ERR_BREAK and 0x00000012: IO_OPEN_CNX_ERROR_BREAK. Added 0x0000003F IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY. Section 8.10 Added OSSA_IO_XFER_ERROR_SATA error and 0x0000000E: IO_XFER_ERROR_BREAK. Section 8.11 Changed 0x0000000E: IO_XFER_ERROR_BREAK. Tables 122 SSP_COMPLETION Response Fields, Table 124 SSP_COMPLETION Response Formats and Table 134 SATA_COMPLETION Response Fields: Corrected Buffer Count size from 4 to 5 bits. <p>Section 9:</p> <ul style="list-style-type: none"> Section 9.2.3 Table 186 changed REVID reset state from 0x04 to 0x05; MSI-X Capability reset state changed from 0x000FC011 to 0x000F0011. Section 9.2.44 MASI-X capability Register Reset State changed from 0x000FC011 to 0x000F0011. Table 204 bit [15:8] changed

Issue No.	Issue Date	Details of Change
		<p>from 0xC0 to 0x00.</p> <p>Section 10:</p> <ul style="list-style-type: none"> • Section 10.3.1 Corrected Boot Strapping Bit Register defaults. • Section 10.3.13 Inserted Inbound AXI Translation Upper Address Window 2 Register • Section 10.3.7 SPC_RESET Table 228 bit [31] added set 1 case • Section 10.5.12 Table 268 bit 26 changed to bit 24 • Table 269 SAS/SATA PHY Layer Address Map: Added Maximum AIP Allowed and Open Retry Interval • Section 10.6.13 Table 282 bit [19:18] and [17:16] set default value to be 00 • Section 10.9.1 Table 316 Title changed from status register to enable register. Bit [31:0] changed to 0000 0000 0000 1000 0000 0000 0001 0000 • Section 10.9.1 Table 317 bit 19 changed to RB 6 for both AAP1 and IOP • Section 10.9.2 Table 318 Title changed from status register to enable register • Various changes in register formats per customer <p>Section 11:</p> <ul style="list-style-type: none"> • Section 11.2.2.5 Table 342 MBIC - INTERNAL_SET1_ERR Details removed bit 3 to 0. • Section 11.2.15 Fig 50 and Table 354 Register dump update with more registers available when fatal error occurs. • Section 11.6.2.5 Table 357 Abandon Class OPEN_REJECT Primitives - Added IO_OPEN_CNX_ERROR_STP_RESOURCES_BUSY
7	May 2009	<p>Section 3:</p> <ul style="list-style-type: none"> • Updated the event log header description in section 3.17.1. • Corrected the TimeStamp Upper and Lower descriptions in section 3.17.2. • Updated description for bringing the boot ROM into HDA mode using an invalid ILA image and by a Soft Reset in section 3.21.1. <p>Section 5:</p> <ul style="list-style-type: none"> • Added the IOABTDLY description and modified 0x12 description in the MPI Configuration table (table 37) in section 5.2.1. • Updated HMI_ERR in section 5.2.2 General Status Table. • Added additional offsets to step 3 of section 5.2.6.1 Host-SPC 8x6G MPI Initialization. <p>Section 7:</p> <ul style="list-style-type: none"> • Updated GPIOEVFALLdescription in section 7.21, GPIO Command. Rearranged location of the DW2 bits in the table (Event OBID, GPIO Event Setup, GPIO Pin Setup, GPIO Read, GPIO Write.

Issue No.	Issue Date	Details of Change
		<p>Section 8:</p> <ul style="list-style-type: none"> • Added 0x0000003F description to the STATUS field in section 8.4 SMP_COMPLETION Response. • Updated GPIOEVFALL description in section 8.16, GPIO Response. • Updated the STATUS description of the SSP_ABORT Response in section 8.19. • Changed DWord 3[15:0] Length from 4 Bytes to 2 Bytes in Table 172, section 8.29. <p>Section 9:</p> <ul style="list-style-type: none"> • Section 9.2.3 Table 186 Reset State value changed for registers with following offset address in configuration space: 0x20, 0x42, 0x71, 0x72, 0x74, 0x78, 0x7C, 0xAC, 0xB0, 0xB4, 0x10C. Detailed information is listed in individual register change. • Section 9.2.27 Reset State value changed from 0x0BC2 to 0x0BC3. Table 191 bit [2:0] changed from 0x02 to 0x3. • Section 9.2.31 Reset State value changed from 0x0080 to 0x008A. Table 193 bit [3:1] changed from 0x0 to 0x5. • Section 9.2.36 Reset State value changed from 0xB0 to 0xAC. • Section 9.2.37 Reset State value changed from 0x0001 to 0x0002. Table 196 bit [3:0] changed from 0x1 to 0x2. • Section 9.2.38 Reset State value changed from 0x00008241 to 0x00008701. Table 197 bit [11:9] changed from 0x1 to 0x3, bit [8:6] from 0x0 to 0x4. • Section 9.2.39 Reset State value changed from 0x2010 to 0x2810. Table 198 bit [11] from 0x0 to 0x1. • Section 9.2.41 Reset State value changed from 0x00133C82 to 0x00033C82. Table 200 bit [20] from 0x1 to 0x0. • Section 9.2.44 Reset State value changed from 0x000F0011 to 0x000FC011. Table 203 bit [15:8] from 0x00 to 0xC0. • Section 9.2.45 Reset State value changed from 0x001C2000 to 0x00002000. Table 204 bit [31:3] from 0x38400 to 0x400. • Section 9.2.46 Reset State value changed from 0x001C4000 to 0x00004000. Table 205 bit [31:3] from 0x38800 to 0x800. • Section 9.2.50 Reset State value changed from 0x00006210 to 0x00062030. Table 209 bit [5] from 0x0 to 0x1. <p>Section 10:</p> <ul style="list-style-type: none"> • Section 10.3.1 TOP_BOOT_STRAP changed Table 221 bit [26:0] from 0x0 to 0x3E0A682. • Section 10.3.3 TOP_DEVICE_REV changed Table 223 bit [3:0] from 0001 to 0010. • Section 10.3.4 TOP_SAS_LOCK_DET_STATUS changed Table 224 bit [1] from 0 to 1, bit [0] from 0 to 1. • Section 10.3.7 SPC_RESET changed Table 227 bit [16] from 1 to 0. • Section 10.4.2 TRS0_CTRL changed Table 238 bit [26:24] from 000 to 100 and added 010, 011, 100, 101, 110, 111 code

Issue No.	Issue Date	Details of Change
		<p>descriptions. Added TRS0_MODE for bit [15:0] with default value to 0000 0000 0000 0001.</p> <ul style="list-style-type: none"> • Section 10.5.1 CONTROL_1 changed Table 256 all bit to be 0. • Section 10.5.2 TIMER_0 changed Table 257 bit [31:16] to be 0x0020 and bit [15:0] to be 0x0020. • Section 10.5.4 Mnemonic name REG12 changed to TX_PPC_SAS_SATA_G1, changed Table 259 bit [27:26] from 00 to 01. • Section 10.5.5 Mnemonic name REG13 changed to TX_PPC_SAS_SATA_G1, changed Table 260 bit [27:26] from 00 to 01. • Section 10.5.6 Mnemonic name REG14 changed to TX_PPC_SAS_SATA_G3. • Section 10.5.7 Mnemonic name REG14 changed to RX_PPC_SAS_SATA_G1G2, changed Table 262 bit [24:21] and bit [8:5] from 1001 to 1011, bit [20:17] and bit [4:1] from 0111 to 1010. • Section 10.5.8 Mnemonic name REG15 changed to RX_PPC_SAS_SATA_G3, changed Table 263 bit [24:21] and bit [8:5] from 0101 to 1010, bit [20:17] and bit [4:1] from 0101 to 1010. Added name for bit 26, 25, 11, 10, 9. • Section 10.5.9 Mnemonic name REG16 changed to TRS_GLOBAL_CFG. • Section 10.5.10 Mnemonic name REG18 changed to TX_CFG_1. • Section 10.5.11 Mnemonic name REG19 changed to RX_CFG_1. • Section 10.5.12 Mnemonic name REG20 changed to RX_CFG_2, changed Table 267 bit [15:8] to 0000 0010. Added name for bit [15:8]. • Section 10.6.3 Mnemonic name REG10 changed to INT_STATUS_1. • Section 10.6.4 Mnemonic name REG11 changed to INT_VALUE_1. • Section 10.6.6 Mnemonic name REG12 changed to INVAL_DWORD_CNT. • Section 10.6.7 Mnemonic name REG13 changed to DISP_ERR_CNT. • Section 10.6.8 Mnemonic name REG14 changed to CODE_VIOL_ERR_CNT. • Section 10.6.9 Mnemonic name REG15 changed to DWS_LOST_CNT. • Section 10.6.10 Mnemonic name REG16 changed to PHY_RST_FAILD_CNT. • Section 10.6.11 Mnemonic name REG17 changed to PRBS_TEST_PAT_ERR_CNT. • Section 10.6.12 Mnemonic name REG18 changed to SAS2_RX_ERR_CNT. • Section 10.6.13 Mnemonic name REG20 changed to DIAG_CFG. • Section 10.6.14 Mnemonic name REG21 changed to ERR_INS_1.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Section 10.6.15 Mnemonic name REG22 changed to ERR_INS_2. • Section 10.6.16 Mnemonic name REG23 changed to TEST_PAT_INS_1_1. • Section 10.6.17 Mnemonic name REG24 changed to TEST_PAT_INS_1_2. • Section 10.6.18 Mnemonic name REG25 changed to TEST_PAT_INS_2_1. • Section 10.6.19 Mnemonic name REG26 changed to TEST_PAT_INS_2_2. • Section 10.6.20 Mnemonic name REG27 changed to SAS_SETTING_LOCAL, changed Table 288 bit [31] from 1 to 0, bit [1] from 0 to 1. • Section 10.7.5 PERF_2_CONTROL changed Table 294 bit [24:20] from 0 0000 to 0 1111. • Section 10.7.11 TIMER_ENABLES changed Table 300 bit [20:15] from 01 0101 to 11 1111. Changed TIMER_CLK to 100 µs units. • Corrected register reference in Section 10.7.13 MAX_CON_TIMER_VALUE bit description. Changed TIMER_CLK to 100 µs units. • Section 10.8.1 GSM_CFG_AND_RESET changed Table 305 bit [0] from 1 to 0. • Section 10.9.1 NMI_EN_VPE0_AAP1 changed Table 315 bit [31:0] from 0xFFFFFFFF to 0x00080000. • Section 10.9.2 NMI_EN_VPE0_IOP changed Table 317 bit [31:0] from 0xFFFFFFFF to 0x00080000. • Corrected MBIC address map listing in section 10.9 MBIC Registers. • Added section 10.10 miscellaneous registers (specifically GPIO-0 Output Control). <p>Section 11:</p> <ul style="list-style-type: none"> • Updated the firmware general fatal errors description in section 11.2.14. • Updated section 11.2.2.1 GSM Double-Bit ECC Error Details, table 325 GSM – DOUBLE_BIT_ECC_ERR Details. • Updated section 11.2.3.1 MBIC – GSM AXI Error Details, table 331 MBIC - GSM_AXI_ERR Details. • Updated section 11.2.3.2 MBIC – MXCBI AXI Error Details, Table 332 MBIC - MXCBI_AXI_ERR Details. • Updated section 11.2.3.3 MBIC – PCIe AXI Error Details, Table 333 MBIC - PCIE_AXI_ERR Details. • Updated section 11.2.11 ILA Image Loading Error, Table 348 ILA Image Loading Error Details. • Changed IO_UNDERRUN to IO_UNDERFLOW for the Information Unit too short error message described in section 11.6.4.7 and 11.6.5.7. • Updated sections 11.4.1 and 11.4.2 Soft Reset Recovery, Normal and HDA modes, steps 1, 5, 6, and 13.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> Updated the RETRY primitive description in section 11.6.2.5. Updated the Status messages in section 11.7.2.1 CRC Error, 11.7.2.4 SYNC Received During FIS Transmission and the Event message in section 11.7.2.5 R_ERR Received. Updated the Event message in section 11.7.3.3 PIO Data Overflows.
6	February 2009	<p>Section 1:</p> <ul style="list-style-type: none"> Removed references to GPIO. Updated the configurable peripheral interfaces section (section 1.1.5) Removed mux layer from diagram in section 1.4.4 and corrected typo in section 1.4.4.2. <p>Section 2:</p> <ul style="list-style-type: none"> Swapped producer and consumer labels in Figure 4 Circular Queues in section 2.2. Changed ROMBASE description in section 2.6 Memory Address Space. Updated Extended Scatter/Gather list wording in Figure 8. In section 2.1.3, removed MPI reference from the from the list of structures in host memory. <p>Section 3:</p> <ul style="list-style-type: none"> Updated the firmware/partition update section (section 3.15) Updated section 3.16 Firmware/Partition Update, removed the support of 8 bit wide flash memory, and clarified the table fields. Added DS_IN_ERROR device state in section 3.2 Device Handle and DEVICE_ID. Updated section 3.16.1 Flash Image Verification. Changed the Extended bit from 1 to 0 in Figure 34 SATA Host Write Example - Local Gather List and Figure 39 SATA Host Read Example - Local Gather List Changed the HDAILA references in Tables 23 and 24 to ILAHDA. Removed Section 3.10 "SMP Response Operations" since SMP target functionality is not supported. Minor edits to Table 16 in Section 3.19. Updated the timer resolution in Table 15 Event Log Entry Format in section 3.7.2. Corrected typos in titles for Figure 16 and 17. Added section 3.23 Expansion ROM Support. <p>Section 4:</p> <ul style="list-style-type: none"> Changed flow diagram in section 4 Initialization and Configuration. Various pin updates to section 4.2. Several updates to Section 4.6 Flash Memory Partition Format. Edited Section 4.7, PCIe System Configuration. Removed Boot Process Status Indication section.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Hid sections 4.3.1 and 4.3.2 since these details are not relevant to customers. <p>Section 5:</p> <ul style="list-style-type: none"> • Added Forced Normal Priority on All OQs (FNPOQ) option in option in dword 0x1C[7] on section 5.2.1 Main Configuration Table. • Added interrupt re-assertion options in dword 0x1C [31:17] on section 5.2.1 Main Configuration Table. • Marked DWords 0x12 and 0x13 of Table 36 MPI Configuration Table – Main Part in Section 5.2.1 as reserved since SMP target functionality is not supported. • Added in section 5.2.2 MPI General Status Table Field: Inbound Table Error code for exceeded the maximum of 32 High Priority Inbound Queues. • Added clarification in section 5.2.3 MPI Inbound Queue Configuration Table Fields, for the initialization of consumer index. • Added in 5.2.2 MPI General Status Table Fields, status of GPIO input pins [19:0]. • Add clarification in section 5.2.6.3 Host-SPC 8x6G MPI Inbound Freeze and 5.2.6.4 Host-SPC 8x6G MPI Inbound Un-Freeze. <p>Section 7:</p> <ul style="list-style-type: none"> • Changed definition of RDF field in section 7.8 SSP_TGT_IO_START Command. • Added ODS field to section 7.8 SSP_TGT_IO_START Command. • Added ODS field to section 7.9 SSP_TGT_RESPONSE_START Command. • Removed “SMP_RESPONSE Command” section since SMP target functionality is not supported. • Added COMINIT_OOB command in section 7.18 LOCAL_PHY_CONTROL Command. • Revised description in section 7.20 FW_FLASH_UPDATE Command to also allow this IOMB to update the firmware of the Expansion ROM. • Clarification in section 7.21 GPIO Command. Updated OT[12..9] and OT[0..11] fields. • Added clarification in 7.22 SAS_DIAG_MODE_START-END Command. • Clarification in section 7.23 SAS_DIAG_EXECUTE Command. Added PMON details to Dword 6 • Updated section 7.25 GET_TIME_STAMP to indicate the resolution is 8 nsec. • Changed section 7.27 GET_NVMD_DATA Command to include the support for reading back the Expansion ROM image from flash memory. Corrected the IP field description for DWord 2[31]. • Updated the IP field description in section 7.28 SET_NVMD_Data Command.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Updated the IPReserved/D_DATA field in section 7.29 SET_DEVICE_STATE Command. • Added section 7.32 SAS_RE_INITIALIZATION Command IOMB. <p>Section 8:</p> <ul style="list-style-type: none"> • In Section 8.2, rearranged 0x000B: IOP_EVENT_BROADCAST_SES" before "0x000C: IOP_EVENT_PHY_ERR_INBOUND_CRC. • Added the RESC_V, RESC_PAD and RESIDUAL_COUNT fields to section 8.3 SSP_COMPLETION Response. • Added 0x0000001A: IO_XFER_ERROR_ACK_NAK_TIMEOUT to Section 8.3 SSP Completion Response. • Updated 0x00000000: IO_COMPLETED in section 8.3 SSP_COMPLETION Response. • Removed 0x00000006: IO_NOT_VALID from section 8.9 SATA_COMPLETION Response. • Removed 0x00000020: IO_XFER_ERROR_SATA from section 8.9 SATA_COMPLETION Response. • Moved 0x0000003C: IO_XFER_PIO_SETUP_ERROR from section 8.9 SATA_COMPLETION Response to section 8.10 SATA_EVENT Notification. • Removed "SMP_REQUEST RECEIVED Notification" section since SMP target functionality is not supported. • Updated 0x0000001A: IO_XFER_ERROR_ACK_NAK_TIMEOUT in Section 8.11 SSP Event Notification. • Updated 0x00000034: IO_XFER_ERROR_OFFSET_MISMATCH in Section 8.11 SSP Event Notification • Updated the "GPIO Response Fields" table in Section 8.16 for OT[0..11] and OT[12..19]. • Added in section 8.22 SAS_DIAG_EXECUTE Response, an new PHY_INVALID status. • Added SCP field to section 8.26 SMP_ABORT Response. • Updated 0x00000000: IO_COMPLETED and 0x00000006: IO_NOT_VALID of section 8.26 SMP_ABORT Response. • Updated the DS field of section 8.32 GET_DEVICE_STATE Response. • Added section 8.34 SAS_RE_INITIALIZATION Response IOMB. <p>Section 9:</p> <ul style="list-style-type: none"> • Changed the reset state in section 9.2.44 MSI-X Capabilities Register. • Updated the description of section 9.2.19 ROM Base Address Register. <p>Section 10:</p> <ul style="list-style-type: none"> • Updated the default in R_RXMODE_3_0 field of the Receiver Configuration 1 Register Bits in section 10.5.11. <p>Section 11:</p>

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Fixed step 15 of Section 11.4.1 Soft Reset Recovery (Normal Mode) and insert new step after step 15. Updated steps 6 and 13. • In section 11.4.2 Soft Reset Recovery (HDA Mode) insert new step after step 14. Updated steps 6 and 13. • Changed the table in section 11.2.11 ILA Image Loading Error. • Added clarification in section 11.5.6.3 Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Device. • Added content in section 11.5.6.4 Head-Of-Line Blocking for SATA Protocol. • Corrected the SSP_RSTB references to be OSSP_RSTB. • Corrected the reference to MEMBASE-III to be MEMBASE-I in step 15 of Table 351. • Removed references to SMP target functionality in Section 11.6.2.3. • Removed Section 11.6.6.2 Vendor-specific SMP received since SMP target functionality is not supported. • Edits to Section 11.2.14 Firmware General Fatal Errors. <p>Changed ISR to ISTR in sections 3.15.2 and 4.5.</p>
5	November 2008	<p>Updated references list.</p> <p>Removed support of multiplexing in Section 1.1</p> <p>Section 2:</p> <ul style="list-style-type: none"> • In Section 2.6.1, minor edits to clarify example. <p>Section 3:</p> <ul style="list-style-type: none"> • Added the definition of device state to Section 3.2. • Added Section 3.2.3 Combo Initiator and Target Mode DEVICE_ID. • Removed the use of AGR in Section 3.7 SSP Target Write Operations. • Added in Section 3.11 “SSP Initiator Task Management Operation” references to special option related to device state. • (PREP 105150) Updated Section 3.16 to remove the reference to the utility used to generate the binaries. Added reference to the firmware release notes for further information. • Updated Section 3.18 to explain how to ensure all event logs are in host memory. <p>Section 4:</p> <ul style="list-style-type: none"> • Updated Section 4.3 Configuration EEPROM description. • Updated Section 4.5, Figure 47 Flash Memory Map and Table 33 Flash Partitions <p>Section 5:</p> <ul style="list-style-type: none"> • Updated Section 5.2.1 “MPI Main Configuration Table Fields” to enable the option to support up to 64 Inbound Queues and 64 Outbound Queues. • Updated the MELSEV and IELSEV definitions in Section 5.2.1 with the latest event log severity information.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Added SAS PHY Analog Setup Table offset in Section 5.2.1 MPI Main Configuration Table Fields. • Added description in 5.2.3 "MPI Inbound Queue Configuration Table Fields" to indicate that the maximum number of High Priority Inbound Queues is still limited to 32 even if the support of 64 IQs and 64OQs is enabled. • Updated 5.2.5.3 Host-SPC 8x6G MPI Inbound Freeze, maximum poll time is 10 micro seconds. • Added step 8 in 5.2.5.1 Host-SPC 8x6G MPI Initialization, for the support of 64 IQs/64OQs. • Added Section 5.2.5 MPI SAS PHY Analog Setup Table Fields. <p>Section 6:</p> <ul style="list-style-type: none"> • Updated OBID from 5 bits to 6 bits to support up to 64 Outbound Queues. Updated the OBID description. <p>Section 7:</p> <ul style="list-style-type: none"> • Globally changed 1 Byte PHYID fields to 4 bits for Section 7.2, PHY_START Command, Section 7.3, PHY_STOP Command, Section 7.14 SMP_RESPONSE Command, Section 7.19 LOCAL_PHY_CONTROL Command, Section 7.23, SAS_DIAG_MODE_START_END Command, and Section 7.24, SAS_DIAG_EXECUTE Command. • Added option in Section 7.2 PHY_START Command for analog PHY setting. • Added options in Section 7.5 SSP_INI_TM_START Command: to automatically set device state to DS_IN_RECOVERY after sending TM. • Added AWT Flag option in Section 7.7 DEVICE_HANDLE_ACCEPT Command • Added new parameter AWT 'A' flag in Section 7.16 REGISTER_DEVICE Command. • Updated GPIO Type in GPIO Command, Section 7.22. • Added loopback diagrams and description to Section 7.24 SAS_DIAG_EXECUTE. • Added note on Section 7.28 GET_NVMD_DATA Command and Section 7.29 SET_NVMD_DATA Command about host requirement to pass correct parameter for TWI operation. • Updated Section 7.29 SET_NVMD_DATA NVMD field. • In Section 7.38 updated "Getting Register Dump Information from Flash Memory" and added "Getting Event Log Information from Flash Memory". Updated the IP and DOA fields. • Added Section 7.30 SET_DEVICE_STATE Command IOMB. • Added Section 7.31 GET_DEVICE_STATE Command IOMB. • Added Section 7.32 SET_DEVICE_INFO Command IOMB. <p>Section 8:</p> <ul style="list-style-type: none"> • Updated the LR and EVENT fields in Section 8.2 SAS_HW_EVENT Notification. • Added IO_TM_TAG_NOT_FOUND status to Section 8.3

Issue No.	Issue Date	Details of Change
		<p>SSP_COMPLETION Response.</p> <ul style="list-style-type: none"> • Added IO_SSP_EXT_IU_ZERO_LEN_ERROR in Section 8.3 SSP_COMPLETION Response. Updated the SSP TAG field. Various edits to the STATUS field descriptions. • Changed Section 8.3 SSP_COMPLETION Response status IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. It will automatically set the device state to DS_NON_OPERATIONAL. • Added IO_DS_NON_OPERATIONAL and IO_DS_NON_OPERATIONAL status in Section 8.3 SSP_COMPLETION Response, Section 8.4 SMP_COMPLETION Response and Section 8.9 SATA_COMPLETION Response. • Various updates to the STATUS field descriptions in Section 8.4 SMP_COMPLETION Response. • Updated the 1-byte PHYID fields to 4-bits in Section 8.5, LOCAL PHY CONTROL Response and Section 8.23 SAS_DIAG_EXECUTE Response. • Updated the PORT_ID field to 4-bits in Section 8.8, GET_HANDLE_DEVICE Response. • Added IO_XFER PIO_SETUP_ERROR in 8.9 SATA_COMPLETION Response. Various updates to the STATUS field descriptions. • Updated Section 8.10 SATA_EVENT Notification and 8.11 SSP_EVENT Notification. Various updates to the EVENT field descriptions. • Added AWT flag and removed First Burst Size in 8.15 GET_DEVICE_INFO Response. • Added new error code to the STATUS field of Section 8.16, FW_FLASH_UPDATE. • Updated GPIO Type in Section 8.17 GPIO Response. • Updated the IP field in Section 8.29 GET_NVMD_DATA Response • Updated Section 8.30 SET_NVMD_DATA Response STATUS field. • Added SET_DEVICE_STATE Response IOMB to Section 8.32. • Added GET_DEVICE_STATE Response IOMB to Section 8.33. • Added SET_DEVICE_INFO Response IOMB to Section 8.34. <p>Section 9:</p> <ul style="list-style-type: none"> • Corrected MSIX_CAP, TBL_OFFSET, and PBA_OFFSET offsets in Sections 9.2 and 9.3 (Tables 183 and 185). • Updated Section 9.2.34 Message Data Register. <p>Section 10:</p> <ul style="list-style-type: none"> • Corrected MEMBASE typo in Table 219 of Section 10.3 for the SPC Reset Register. • Added Section 10.3.6 Device LCLK Generation Register. • Corrected mnemonic for the Lane N Receiver RX Mode 2 Register in Section 10.4.6. • Added Section 10.5.3, Timer Control 2 Register.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Added Section 10.6.5 Counter Configuration Register. <p>Section 11:</p> <ul style="list-style-type: none"> • Updated Section 11.2.1 Block Direct Memory Access (BDMA) Fatal Errors. • Updated Section 11.2.2.1 GSM Double-Bit ECC Error Details. • Updated Section 11.2.8.2 HSST – BDMA Timer Expired Error Details. • Updated Section 11.2.10 PCIe General Fatal Error. • Updated Section 11.2.11 ILA Image Loading Error. • Updated Section 11.2.12 Firmware Assert Fatal Error. • Updated Section 11.2.13 Firmware Watchdog Timer Fatal Error. • Updated Section 11.2.14 Firmware General Fatal Errors. • Added Section 11.5.6 Device States Related Error Handler. • Various updates to Section 11.6 Detailed Descriptions of the SAS Error Conditions and 11.7 Detailed Description of SATA Error Conditions.
4	September 2008	<p>Updates for Dev 01.06c:</p> <p>Section 4:</p> <ul style="list-style-type: none"> • Changed section title in Section 4.3 from “SEEPROM Configuration” to “Configuration SEEPROM” and other description changes. • Updated Section 4.5, flash memory format. <p>Section 5:</p> <ul style="list-style-type: none"> • Added cross reference to Section 11.2.14 to Section 5.2.1. Updated the IELSEV, MELSEV, MELBS, MELBAH, MELBAL, IELBAH, IELBAL and IELBS fields to clarify event logging configuration. <p>Section 7:</p> <ul style="list-style-type: none"> • Updated DWord 4 to describe ACK NAK retry in Section 7.9, SSP_TGT_RESPONSE_START. • Updated PORT_CONTROL_COMMAND in Section 7.27. Added an option in 7.27 PORT_CONTROL Command for HARD_RESET to do automatic deregistration of devices attached to the port. • Updated NVMD and IPLD fields in Section 7.28 GET_NVMD_DATA Command and Section 7.29 SET_NVMD_DATA Command. <p>Section 8:</p> <ul style="list-style-type: none"> • Removed unsupported error codes, IO_XFER_ERROR_CMD_ISSUE_BREAK_BEFORE_ACK_NAK, IO_LINK_FAILURE, IO_PROG_ERROR, and IO_OPEN_CNX_ERROR_UNKNOWN_ERROR, described in the status field of Section 8.3 SSP_COMPLETION, 8.4 SMP_COMPLETION Notification, 8.9 SATA_COMPLETION Notification, 8.11 SSP_EVENT Notification, and 8.11 SSP_EVENT Notification.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Updated IPDL and STATUS fields in Section 8.29 GET_NVMD_DATA Response and Section 8.30 SET_NVMD_DATA Response respectively. <p>Section 10:</p> <ul style="list-style-type: none"> • Added Section 10.7.2 GSM RAM ECC Double Bit Error Indication register, Section 10.7.6 GSM Read Address Parity Error Indication register, Section 10.7.7 GSM Write Address Parity Error Indication register, and Section 10.7.8 GSM Write Data Parity Error Indication register. <p>Section 11:</p> <ul style="list-style-type: none"> • Added cross reference to Section 11.2. • Added Section 11.2.14 for register dump information during device specific critical error. • Changed title in 11.5.5 from "Target Mode Hard Reset Received Handler" to "Hard Reset Received Handler". • Changed hard reset received handler in Section 11.5.5.1, 11.5.5.2 and 11.5.5.3 to cover both initiator and target. • Removed the cross reference to IO_XFER_ERROR_CMD_ISSUE_BREAK_BEFORE_ACK_NAK from Section 11.6.2.6 BREAK Received and 11.6.2.11 BREAK Received During Connection since this is unsupported.
3	August 2008	<p>Updated for Dev 01.06b:</p> <p>Section 3:</p> <ul style="list-style-type: none"> • Updated Section 3.16, firmware partition update. • Changed cross-references in Section 3.22, Host Direct Access (HDA) Mode. • Clarification of the command and response address in Section 3.22.2, Boot ROM HDA Protocol – Command and Response. • Clarifications on the GSM starting address in Section 3.22.4, Host and SPC 8x6G Initialization Sequence, Table 24. Updated introduction. <p>Section 4:</p> <ul style="list-style-type: none"> • Changed Section 4.2, SPC 8x6G Bootstrap Configuration on LBI_A[25] setting. • Changed Section 4.3.1, General Configuration Table by removing references to hardware default and other updates. • Changed Section 4.5, flash memory format and Section 4.6, flash memory partition format. • Clarification in Section 4.14, Unloading the Host Driver. <p>Section 5:</p> <ul style="list-style-type: none"> • Update description of Scratchpad 1 Register and Scratchpad 2 Register in Section 5.1, MPI Configuration Table Access. • Added GENERAL EVENT Notification Queue in Section, DEVICE_HANDLE_Removed Notification Queue, HDA settings, and other updates to 5.2.1, MPI Main Configuration Table. • Added Outbound Queue n Interrupt Mode field in Section 5.2.4, MPI Outbound Queue Configuration Table Fields. Updated

Issue No.	Issue Date	Details of Change
		<p>Outbound Queue 0 Interrupt Enable and Outbound Queue 0 Interrupt Coalescing Count.</p> <p>Section 7:</p> <ul style="list-style-type: none"> Changed in Section 7.9 SSP_TGT_RESPONSE_START Command. Changed Section 7.15, SMP_ABORT Command parameters. Removed First Burst Size and clarify UPPER DEVICE_ID in Section 7.16, REGISTER_DEVICE Command. Added information in Section 7.21, FW_FLASH_UPDATE Command about the minimum 28 bytes length. Added SET_PORT_RESET_TIME and HARD_RESET in Section 7.27, PORT_CONTROL Command. Updated Section 7.24, SAS_DIAG_EXECUTE. Table labeled SAS Diagnostic Command Type and Command Description Valid Combination. Changed description in 7.25, SAS_HW_EVENT_ACK Command to include IOP_EVENT_PORT_RECOVERY_TIMER_TMO and IOP_EVENT_PORT_RESET_TIMER_TMO. Added Section 7.29, SET_NVMB_DATA. Updated 7.27, PORT_CONTROL Command. Added new GET_NVMD_DATA Command IOMB in Section 7.28, which replaces GET_VPD Command IOMB and TWI Command IOMB. Added the SET_NVMD_DATA Command in Section 7.29, which replaces the SET_VPD Command. Removed GET_VPD Command, TWI Command, and SET_VPD Command. <p>Section 8:</p> <ul style="list-style-type: none"> Added in Section 8.2 SAS_HW_EVENT Notification new events: IOP_EVENT_BROADCAST_ASYNCNCH_EVENT, IOP_EVENT_PORT_RESET_TIMER_TMO and IOP_EVENT_PORT_RESET_COMPLETE. Clarified Section 8.2 SAS_HW_EVENT Notification the description for IOP_EVENT_SAS_PHY_UP, IOP_EVENT_SATA_PHY_UP, IOP_EVENT_PHY_DOWN and IOP_EVENT_HARD_RESET_RECEIVED. Added error status IO_PORT_IN_RESET in Section 8.3 SSP_COMPLETION Response, 8.4 SMP_COMPLETION Response, 8.9 SATA_COMPLETION Response. Added more STATUS in Section 8.6 DEVICE_REGISTRATION Response. Corrected typo in the HTAG field of the DEVICE_REGISTRATION Response format diagram. Fixed the DEVICE_ID field description in Section 8.9 GET_DEVICE_HANDLE Response. Updates to Section 8.12, DEVICE_HANDLE_ARRIVED. Changes in Section 8.19 GENERAL_EVENT Notification, added INBOUND_IOMB_INVALID_OBID status. Updated Section 8.28, SMP_ABORT

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Added new GET_NVMD_DATA Response in Section 8.29, which replaced GET_VPD Response IOMB and TWI Response IOMB. • Added the new SET_NVMD_DATA Response in Section 8.30. • Added the DEVICE_HANDLE_REMOVED Notification to Section 8.31. • Removed IO_XFER_ERROR_ACK_NAK_TIMEOUT from SATA_EVENT and SATA_COMPLETION. • Removed SET_VPD and TWI_RESPONSE. <p>Section 9:</p> <ul style="list-style-type: none"> • Updated the Minor field default of the Revision Register in Section 9.2.8. • Changed Section 9.2.44 MSI-X Capability Register offset and TBL_SZ reset state. • Changed Section 9.2.45 Table Offset Register offset. • Changed Section 9.2.46 PBA Offset Register offset. <p>Section 10:</p> <ul style="list-style-type: none"> • Added Section 10.4.1 OSSP Control 1 Register. • Removed Section 10.4.11 Edge Rate Mode Register as the correct per-PHY register is already described in Section 10.4.8. • Updated the UNPLUG_DET_V, HOLD_DET_V, and RATE_SNOOP_DONE_V in Section 10.5.3. • Updated the UNPLUG_DET_VAL and HOLD_DET_VAL bit descriptions in Section 10.5.4, SSPL Interrupt Values register. • Added Section 10.5.19 SAS2 Settings (Local) Register. <p>Section 11:</p> <ul style="list-style-type: none"> • Updated step 1 in the Soft Reset Sequence (Normal Mode) and Soft Reset Sequence (HDA Mode) in Section 11.4. • Various updates to Section 11.5.1, PHY Down Handler. • Added Section 11.5.2 Local PHY Control Link/Hard Reset Handler. • Added Section 11.5.5 Target Mode Hard Reset Received Handler. • Corrected description in Section 11.6.4.6. • Removed various references to DDR. Not applicable to this product.
2	June 2008	<p>Updated for Dev 01.06a:</p> <p>Section 1</p> <ul style="list-style-type: none"> • Updated Section 1.2.1. <p>Section 2:</p> <ul style="list-style-type: none"> • Added new Section 2.6 to describe the conventions for using the Memory Address space to include the new 64-KB MEMBASE mode details. • Moved the “Default PCIe BAR Configuration per Detected Device Type” table to Section 2.6. Updated the table for the 64-KB mode. • Removed the ROMBASE AXI address from Section 2.6.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Added the inbound window shifting procedure to Section 2.6.1. <p>Section 3:</p> <ul style="list-style-type: none"> • Section 3.1 updated cross reference to PHY Initialization. • Changed Section 3.2 Device Handle and DEVICE_ID: removing references to auto registration for directly attached device. • Changed Section 3.3 SAS Discovery, directly attached device will not be automatically registered. • Updated direction of data transfer description in Sections 3.5 and 3.6. • Updated Section 3.8.1 to correct the host configuration of the DIR field. • Changed in Section 3.18.2 Event Log Entry, time stamp is now 64-bit and common to both MSGU and IOP. • Updated Section 3.19 to add high priority details. • Added new Section 3.22 Host Direct Access (HDA) Mode. Removed normal MEMBASE mode details. • Added new Section 3.23 SMP Request Serialization. <p>Section 4:</p> <ul style="list-style-type: none"> • Added HDA mode to bootstrap configuration details in Section 4.2. • Moved the “Default PCIe BAR Configuration per Detected Device Type” table to Section 2. • Modification in Section 4.3.1 General Configuration Table, byte 4 bit 5 is defined for HDA mode selection. • Changes in Section 4.9 Boot Process Status Indication bit definition. • Modification in Section 4.11 SPC Reset and 4.12 Host Initialization, to include HDA mode. • Updated Section 4.11 for time delay during a hard power-on reset to 300 msec. • Added Section 4.14 Unloading Host Driver. <p>Section 5:</p> <ul style="list-style-type: none"> • Added in Section 5.2.1 MPI Main Configuration Table Field, fields for MSGU and IOP register dumps. • Changed Section 5.2.1 MPI Main Configuration Table Field, added Number of Devices (MD) field and made Maximum Scatter-Gather List Elements (MSGLE) a 16-bit field. Made sure the SAS revision specification is for SPC 8x6G. • Changed Section 5.2.1 MPI Main Configuration Table Field, for definition of MSGU Event Log Severity. • Various changes in Section 5.2.2 MPI General Status Table Fields. • Updated write access to 0x08 in step 3 of Section 5.2.5.1. • Changed Section 5.2.5.3 Host-SPC 8x6G MPI Inbound Freeze, the purpose of freezing IQ. • Changed Section 5.2.5.2 Host-SPC 8x6G MPI Communication Termination sequence.

Issue No.	Issue Date	Details of Change
		<p>Section 6:</p> <ul style="list-style-type: none"> Added H bit high priority flag in IOMB header. <p>Section 7:</p> <ul style="list-style-type: none"> Added the H bit high priority flag (described in Section 6) to all the Command header format diagrams. Added commands to Summary of Inbound Operations Codes table. Updated Section 7.3 PHY_START Command (spin-up) Changed Section 7.4 PHY_STOP Command, no PHY down will be reported. Clarified DIR field in Section 7.5 and 7.7. Updated the SSPIU field of Section 7.7 SSP_INI_EXT_IO_START Command. Added Message Report 'M' field option in Section 7.5 SSP_INI_IO_START Command, Section 7.6 SSP_INI_TM_START Command, Section 7.7 SSP_INI_EXT_IO_START Command and 7.18 SATA_HOST_IO_START Command, to allow event notification when frame has been sent on the wire. Changed Section 7.8 DEVICE_HANDLE_ACCEPT Command to allow host to assign the upper 16 bit of DEVICE_ID. Named the Host Assigned Upper DEVICE_ID field HA. Changed Section 7.9 SSP_TGT_IO_START Command fields' position and definition. Updated the DirectM field of SSP_INI_IO_START and SSP_INI_EXT_IO_START in Section 7.6 and 7.18. Added direct payload mode in Section 7.10 SSP_TGT_RESPONSE_START Command. Updated other fields. Changed Section 7.11 SSP_ABORT Command for abort per device response status. Changed Section 7.12 Deregister_Device_Handle Command: remove restriction for deregistering directly attached device. Changed in Section 7.13 GET_DEVICE_HANDLE Command, removed DEVA field and changed DEVT definition. Fixed typo in Section 7.14 SMP_REQUEST in the Indirect SMP Request Frame Length. Changed Section 7.16 SMP_ABORT Command description. Changed Section 7.17 REGISTER_DEVICE Command: added support for directly attached SATA device registration and host assigned of lower 16-bits of DEVICE_ID. Other minor edits. Added note, updated R bit description. Named the Host Assigned Upper DEVICE_ID field HA. Updated description for SATA_HOST_IO_START Command in Section 7.18. Changed Section 7.19 SATA_ABORT Command for abort per device response status.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Section 7.21 minor edits. • Updated the Current Image Length (CLEN) in Section 7.23, FW_FLASH_UPDATE Command Field. • Updated the VPD Device (VPDD) field in Section 7.24. • Changed Section 7.25 GPIO Command to include OBID for GPIO event. • Removed IN_CONN_CRC_ERR_CNT and CRC_ERR_CNT from SAS Diagnostics and Command Type and Command Descriptors Valid Combinations table and the Command Descriptors for Command Types DIAG_REPORT_GET and ERR_CNT_RESET table in Section 7.27. Added SSPA_PERF_CNT and PHY_RST_CNT. • Updated Section 7.27 SAS_DIAG_EXECUTE Command description. Added PERF1CTL to the SAS_DIAG_EXECUTE format and field descriptions. Added PHY_RST_CNT, SSPA_PERF_CNT, and removed IN_CONN_CRC_ERR_CNT from SAS Diagnostic Command Type and Command Description Valid Combination and Command Descriptors for Command Types DIAG_REPORT_GET and ERR_CNT_RESET tables. Added "Get error count of the SSPA Performance Count register" to the SAS Diagnostic Command — Required Argument table. • Added Section 7.28 SAS_HW_EVENT_ACK Command IOMB. Added 0x00C to the SEA field. • Added Section 7.29 GET_TIME_STAMP Command IOMB. • Added Section 7.30 PORT_CONTROL Command IOMB. <p>Section 8:</p> <ul style="list-style-type: none"> • Added the H bit high priority flag (described in Section 6) to all the Command header format diagrams. • Added outbound operations codes to summary table. • Changed Section 8.3 SAS_HW Event, renamed IOP_EVENT_INBOUND_CRC_ERROR to IOP_EVENT_PHY_ERR_INBOUND_CRC. Removed IOP_EVENT_LOSS_OF_SIGNAL. • Changed Section 8.3 SAS_HW Event, removed the auto registration of directly attached device, the DEVICE_ID field is removed. Removed IOP_EVENT_PORT_INVALID event and added PS (Port State) and NPIP (Number of Phys In Port) fields. Changed description of BROADCAST CHANGE, PHY down, PHY error events and IOP_EVENT_PHY_ERR_INBOUND_CRC. Added a new event IOP_EVENT_HARD_RESET_RECEIVED, IOP_EVENT_PHY_DOWN and IOP_EVENT_INBOUND_CRC_ERROR. Updated SSP Completion Status. Added IOP_EVENT_PORT_RECOVER and IOP_EVENT_PORT_RECOVERY_TIMER_TMO. • Changed Section 8.4 SSP_COMPLETION Response, added SSP TAG field. Clarified IO_NO_DEVICE. Clarified the description of PARAM. • Updated Section 8.5 SMP_COMPLETION. Clarified PARAM description. Added IO_OVERFLOW, IO_ABORTED, AND IO_ERROR_SMP_RESOURCE description.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Removed IO_FAILED status from Section 8.4 SSP_COMPLETION Response, Section 8.5 SMP_COMPLETION Response, and Section 8.10 SATA_COMPLETION Response. • Removed IO_ABORT_RESET from Section 8.4 SSP_COMPLETION Response, Section 8.5 SMP_COMPLETION Response, and Section 8.10 SATA_COMPLETION Response. • Added in 8.4 SAS_HW_EVENT Notification new events IOP_EVENT_PORT_RECOVER and IOP_EVENT_RECOVERY_TIMER_TMO and new states. • Changed in Section 8.5 SMP_COMPLETION Response: clarification for the PARAM field and added IO_OVERFLOW status. • Changed Section 8.5 SMP_COMPLETION Response by adding IO_ABORTED and IO_ERROR_INTERNAL_SMP_RESOURCE status. • Updated Section 8.7 DEVICE_REGISTRATION Response with clarification on the lower 16-bit and upper 16-bit of DEVICE_ID. Updated STATUS field description • Changed in Section 8.8 Deregister_Device_Handle Response: the status FAILURE_DEVICE_DIRECT_ATTACH (0x00000003) has been removed as now directly attached device registration and deregistration is host controlled. • Removed IO_OVERFLOW from Section 8.10 SATA_COMPLETION Response. Clarified IO_NO_DEVICE. Updated IO_XFER_ERROR_REJECTED_NCQ_MODE description. • Added in Section 8.11 SATA_EVENT Notification, the event IO_XFER_CMD_FRAME_ISSUED when the "M" field is set in SATA_HOST_IO_START Command. Updated PORT_ID. • Added in Section 8.12 SSP Event a new field for SSP Tag. • Added in Section 8.12 SSP_EVENT Notification, the event IO_XFER_CMD_FRAME_ISSUED when the "M" field is set in SSP_INI_IO_START Command, SSP_INI_TM_START Command and SSP_INI_EXT_IO_START Command. Updated PORT_ID. • Added TLR field in Section 8.15 SSP_REQUEST_RECEIVED Event. • Changed in Section 8.16 GET_DEVICE_INFO Response, changed the 'S' field to two bits and added the Retry 'R' field. Updated GET_DEVICE_INFO Response name. • Changed the IOMB name in Section 8.16 from "DEVICE_INFO Response" to "GET_DEVICE_INFO Response". • Updated the Flash Update Status field in Section 8.17, FW_FLASH_UPDATE Response. • Updated the Response Status field in Section 8.22, TWI Response. • Changed the STATUS definition in 8.23 SSP_ABORT Response. Updated SCP. • Changed the STATUS definition in 8.24 SATA_ABORT

Issue No.	Issue Date	Details of Change
		<p>Response. Updated SCP.</p> <ul style="list-style-type: none"> • Added Section 8.27 GET_TIME_STAMP Response. • Added Section 8.28 SAS_HW_EVENT_ACK Response. • Added in Section 8.29 PORT_CONTROL Response IOMB. Updated PORT_ID. • Added Section 8.30 SKIP_ENTRIES_EVENT Notification. • Added Section 8.31 SMP_ABORT Response. <p>Section 9:</p> <ul style="list-style-type: none"> • Updated Section 9.2.1 and 9.2.3 to include MEMBASE details for 64-KB MEMBASE mode. • Updated the Memory Base Address I register description in Section 9.2.14 for 64-KB MEMBASE mode. • Updated the Memory Base Address II register description in Section 9.2.15 for 64-KB MEMBASE mode. • Removed the I/O Base Address register description in Section 9.2.16. • Updated the Memory Base Address III register description in Section 9.2.17 for 64-KB MEMBASE mode. • Created the Memory Base Address IV register description in Section 9.2.18 for 64-KB MEMBASE mode. <p>Section 10:</p> <ul style="list-style-type: none"> • Updated Section 10 to include details about 64-KB MEMBASE mode. This includes updates to the start of the chapter, the subsection introductions and the addresses of all the chip registers for 64-KB MEMBASE mode. • Changed Section 10.1.5 Scratchpad 0 description to include HDA support. • Fixed Section 10.1.6 Scratchpad 1 Register bit definition ([31:8], [7:3]). Added CPU_SOFT_RESET_RDY bit description. • Added CPU_SOFT_RESET_RDY and HOST_SOFT_RESET_RDY bit descriptions to Section 10.1.7 Scratchpad 2 Register. • Changed Section 10.1.9 Host Scratchpad 0 to include both Soft Reset and Soft Reset (HDA Mode) signatures. • Added description to Section 10.1.12 Host Scratchpad 3 for HDA initialization sequence. • Updated Section 10.2.2 DEVICE_ID reset state to 0x8001. • Corrected DEVICE_REVISION default in Section 10.2.3. • Changed Section 10.2.6 SPC Reset Register bit definition. Added reserved bits. • Added Section 10.2.11 Inbound AXI Translation Lower Address – Window 2 register description. • Added register descriptions to Sections 10.2.7, 10.2.8, 10.2.9, and 10.2.10 for the PCIe event and error interrupts. • Added Section introductions to Sections 10.3-10.8. • Various updates to the PCIe analog configuration registers:

Issue No.	Issue Date	Details of Change
		<p>Section 10.3.2 Lane N TRS Control Register, Section 10.3.3 Lane N Transmitter Mode Register, Section 10.3.4 Lane N Transmitter Control Register, Section 10.3.5 Lane N Receiver RX Mode 1 Register, Section 10.3.6 Lane N Receiver RX Mode 2 Register, and Section 10.3.7 Lane N Receiver Status Register.</p> <ul style="list-style-type: none"> Updated the SMP_MAX_CONN_TIMER description in the Timer Enables Register in Section 10.6.11. Added Section 10.4.1 Timer Control 0 Register. Added Section 10.4.2 Transmitter Per Port Configuration 1 SAS/SATA G1 Register and Section 10.4.3 Transmitter Per Port Configuration 1 SAS/SATA G2 Register. Various updates to Section 10.4.4 Transmitter Per Port Configuration 1 SAS/SATA G3 Register, Section 10.4.5 Receiver Per Port Configuration 1 SAS/SATA G1G2 Register, Section 10.4.6 Receiver Per Port Configuration 1 SAS/SATA G3 Register, Section 10.4.7 Global Configuration 1 Register, Section 10.4.8 Transmitter Configuration 1 Register, and Section 10.4.9 Receiver Configuration 1 Register. Removed Section 10.4.8 DFE Configuration 1 Register (offset 0x20150) and Section 10.4.10 LOS Mode Register (offset 0x20904) since these are no longer relevant for SPC 8x6G Rev B and C. Added Section 10.4.10 Receiver Configuration 2 Register. Added the RB6 Access Register definition to Section 10.7.5. Added the MBIC register descriptions to Section 10.8.1 and 10.8.2. <p>Section 11:</p> <ul style="list-style-type: none"> Corrected bit range for source of the fatal error Added in Section 11.2 Device Specific Fatal Errors, to include references to Section 5.2 that describes location for internal registers dump. Changed Section 11.2 Device Specific Fatal Errors, replaced FW_INIT_ERROR to ILA_FWLD_ERR. Removed SSPA_ERR, SSPL_ERR, PCIE_AL_ERR, and PCIE_PCS_ERR from Table 287 and 288. Removed SSPA and SSPL from the list of errors. Updated Table 287 and 288 with CPU_SOFT_RESET_RDY and HOST_SOFT_RESET_RDY Changed table in Section 11.2.7.1 OSSP – GSM Decode Response Parity Error Details. Changed table in Section 11.2.7.2 OSSP – FIFO ECC Error Details. Removed Section 11.2.8 and 11.2.9. Changed in Section 11.2.12 ILA Image Loading Error. Changed 11.2.12 Firmware Initialization Fatal Error Changed 11.2.13 FW Assert Error Details. Added Soft Reset HDA in Section 11.4. Distinguished between Normal mode and HDA mode. Removed normal MEMBASE mode details. Added new step 1 to Section 11.4.1 and 11.4.2 Changed Soft Reset Sequence step 2 in Section 11.4.1 to clear the PCIe

Issue No.	Issue Date	Details of Change
		<p>events and errors and add NMI details. Updated step 15.</p> <ul style="list-style-type: none"> Corrected link with PHY. Added introduction. Added Section 11.5.1 PHY DOWN Handler and subsections. Added Section 11.5.2 BROADCAST CHANGE Handler. Added Section 11.5.3 PHY ERROR Handler. Updated Section 11.6.2.3 Invalid SMP Frame to replace IO_FAILED with IO_XFER_ERROR_RX_FRAME. Changed Section 11.6.2.1 SAS_HW Event, renamed IOP_EVENT_INBOUND_CRC_ERROR to IOP_EVENT_PHY_ERR_INBOUND_CRC. <p>Global</p> <ul style="list-style-type: none"> Changed AAP references to AAP1.
1	March 2008	<p>Document created. The following changes have been made from PMC-2070028:</p> <p>Rebranded document to the PM8001.</p> <p>Section 1:</p> <ul style="list-style-type: none"> Removed the AES/DIF details. Only relevant to PM8002. Updated power specification in features list. <p>Section 2:</p> <ul style="list-style-type: none"> Fixed SATA Endianness description in Section 2.1.3.2, SATA Payload Endianness. <p>Section 3:</p> <ul style="list-style-type: none"> Removed EDC Section. Only relevant to PM8002 Updated Section 3.16.2 Firmware Update Logic. Added Section 3.19 High Priority Operation and Normal Priority IOMB Processing Added Section 3.20 Host Interrupt Usage Model Removed firmware assisted discovery. Also removed related IOMBs in Section 7 and 8. Added Section 3.21 Transport Layer Retry (TLR) Handling. Removed assisted discovery reference in Section 3.2.1 <p>Section 4:</p> <ul style="list-style-type: none"> Updated Section 4.3, EEPROM configuration. <p>Section 5:</p> <ul style="list-style-type: none"> Changed the reading/writing sequence of Configuration Table in Section 5.2.5.1, 5.2.5.2, 5.2.5.3 and 5.2.5.4 for rev B chip. Changed in Section 5.2.1 MPI Configuration Table – Main Part, to allow mapping from PHY_ID to OQ. Also changed DWord 9. Removed Outbound Queue Interrupt Enable field from MPI Outbound Queue Configuration Table. The masking and unmasking of interrupt is host controlled through ODMR register. Changed maximum controller ready indication wait or polling time in Section 5.2.5.1 from 5 seconds to 1 second.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Updated Outbound Queue 0 Interrupt Coalescing Timeout and Outbound Queue n Interrupt Coalescing Timeout. <p>Section 6:</p> <ul style="list-style-type: none"> • Updated Table 31 IOMB Command Header Fields "OBID". <p>Section 7:</p> <ul style="list-style-type: none"> • Clarified PHY_START command spin up hold is for directly-attached SATA devices. • Added SKIP COUNT. • Remove GET_INFO Command as info is available in Configuration Table and PCIe Register. • Changed field definition in Section 7.6 SSP_INI_TM_START Command. • Changed Section 7.9 SSP_TGT_IO_START Command • Removed SSP_INI_EDC_IO_START. Changed in Section 7.11 SSP_INI_EDC_IO_START Command. • Removed SSP_INI_EDC_EXT_IO_START. Changed in Section 7.12 SSP_INI_EDC_EXT_IO_START Command • Removed SSP_TGT_EDC_IO_START. Changed in Section 7.13 SSP_TGT_EDC_IO_START Command • Updated Section 7.11 SSP_ABORT Command descriptions • Added description in Section 7.12 DEREGISTER_DEVICE_HANDLE Command for restriction on directly attached device. • Changes in Section 7.13 GET_DEVICE_HANDLE Command for maximum device handles requested. • Removed PORT_ID field from Section 7.16 DEREGISTER_DEVICE_HANDLE Command. • Removed assisted discovery reference in Section 7.13 • Removed PHY Override in Section 7.14 SMP_REQUEST Command. • Added DEVICE_ID field in Section 7.16 SMP_ABORT Command. • Removed ASSISTED_DISCOVERY Command. • Removed SASDT field from Section 7.17 REGISTER_DEVICE Command. • Added BROADCAST ASYNCHRONOUS EVENT operation in Section 7.20 LOCAL_PHY_CONTROL Command. • Changed Section 7.27 SAS_DIAG_EXECUTE Command valid, Command Type, Command Description, THRSHLD and PMON. • Updated Section 7.19, SATA_ABORT Command descriptions <p>Section 8:</p> <ul style="list-style-type: none"> • Removed GET_INFO Response. • Changed description in Section 8.3 SAS_HW Event for event IOP_EVENT_PORT_INVALID.

Issue No.	Issue Date	Details of Change
		<ul style="list-style-type: none"> • Added new status FAILURE DEVICE ALREADY REGISTERED to Section 8.7 DEVICE REGISTRATION Response. • Changed description of 8.9 GET_DEVICE_HANDLE Response for maximum device handles in the response. • Clarified Big Endian format for SAS address fields in Section 8.16 DEVICE_INFO Response. • Section 8.3 SAS_HW Event changed the PORT_ID validity condition for events. • Update STATUS field in Section 8.8 DEREGISTER_DEVICE_HANDLE Response for failed cases. • Changed PARAM field definition for SMP response length in Section 8.5 SMP_COMPLETION Response. • Changed in Section 8.4 SSP_COMPLETION Response, event IO_XFER_ERROR_CREDIT_TIMEOUT has been merged with event IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. • Added in Section 8.10 SATA_COMPLETION Response STATUS IO_XFER_ERROR_NAK_RECEIVED and IO_XFER_ERROR_ACK_NAK_TIMEOUT. Updated IO_XFER_ERROR_REJECTED_NCQ_MODE. • Added in Section 8.11 SATA Event EVENT IO_XFER_ERROR_NAK_RECEIVED, IO_XFER_ERROR_ACK_NAK_TIMEOUT, and IO_XFER_ERROR_PEER_ABORTED. • Added in Section 8.12 SSP Event additional events: IO_XFER_ERROR_UNEXPECTED_PHASE, IO_XFER_ERROR_XFER_RDY_OVERRUN , and IO_XFER_ERROR_XFER_RDY_NOT_EXPECTED. • Changed in Section 8.15 SSP_REQUEST RECEIVED Event. • Replaced IOP_EVENT_LINK_ERROR in Section 8.3 SAS_HW Event with multiple distinct link error events. • Changed the hex number assignment of IO_XFER_ERROR_ABORTED_NCQ_MODE in Section 8.11 SATA_EVENT Notification from 0x00000021 to 0x00000023. • Added IR field in Section 8.2 GET_VPD Response. • Added the IO_XFER_OPEN_RETRY_TIMEOUT event description to the SMP_COMPLETION Response, SATA_EVENT Notification, and SSP_EVENT Notification. • Updated the SSP_ABORT, SATA_ABORT Response descriptions. • Updated GENERA_EVENT notification (Section 8.21): STATUS. <p>Section 9:</p> <ul style="list-style-type: none"> • Fixed bits description in MEMBASE-I, MEMBASE-II, and MEMBASE-III. • Updated Power Management Control and Status Register "Power State". • Updated DEVICE_ID reset to 0x8001.

Issue No.	Issue Date	Details of Change
		<p>Section 10:</p> <ul style="list-style-type: none"> • Changes in MSGU register description in Section 10.1 to reflect chip rev B. • Updated DEVICE_ID reset to 0x8001. • Added the Outbound Doorbell Mask Register. • Added Section 10.2.7 Message Unit Outbound Doorbell Auto Clear register. • Added Section 10.2.8 Interrupt Coalescing Timer Register. • Added Section 10.2.9 Interrupt Coalescing Control Register. • Updated the access privilege for the Scratchpad registers (Section 10.1.5 to 10.1.8) • Added the GSM registers. <p>Section 11:</p> <ul style="list-style-type: none"> • Updated Section 11.2 Device Specific Fatal Errors (Scratchpad 1 and Scratchpad 2 registers and Firmware Watchdog Timer Fatal Error (Section 11.2.14. Fixed typo on 11.2 Device Specific Fatal Error, on the use of Scratchpad 3 and Scratchpad 2 registers. Added Section 11.2.15. • Changes in Section 11.6 Detailed Descriptions of the SAS Error Conditions. • Changed in Section 11.4.1 Soft reset Recovery sequence steps. <p>Globally replaced IOP_EVENT_SAS_LINK_UP with IOP_EVENT_SAS_PHY_UP, IOP_EVENT_SATA_LINK_UP with IOP_EVENT_SATA_PHY_UP, IOP_EVENT_LINK_ERROR_XX with IOP_EVENT_PHY_ERROR_XX, and IOP_EVENT_LINK_DOWN with IOP_EVENT_PHY_DOWN</p>

Table of Contents

Legal Information.....	2
Contacting PMC-Sierra.....	3
Revision History.....	4
Preface	60
1 Introduction	62
1.1 Features	62
1.1.1 SAS/SATA Interface	62
1.1.2 PCI Express (PCIe) Port	62
1.1.3 Statistics and Performance Monitoring	63
1.1.4 High Speed I/O.....	63
1.1.5 Configurable Peripheral Interfaces	63
1.1.6 Physical	63
1.2 SPC 8x6G Applications	63
1.2.1 Fabric-Attached SAS/SATA Storage System	63
1.3 SPC 8x6G Block Diagram.....	65
1.4 Block Summary	65
1.4.1 Block Direct Memory Access (BDMA).....	66
1.4.2 Global Shared Memory (GSM)	66
1.4.3 Processor Complex System (PCS)	66
1.4.4 Octal SAS/SATA Processor (OSSP) Subsystem	67
1.5 Inbound Queue (IQ)	68
1.6 Outbound Queue (OQ).....	68
2 General Programming.....	69
2.1 Endianness.....	69
2.1.1 Big Endian Format	69
2.1.2 Little Endian Format.....	69
2.1.3 Host Memory Endianness	70
2.2 Circular Queues	70
2.3 Inbound Queues (IQs).....	72
2.3.1 IQ Attributes	73
2.3.2 IQ Producer and Consumer Indexes	73
2.4 Outbound Queues (OQs)	74
2.4.1 OQ Attributes.....	74

2.5	Scatter/Gather Lists (SGLs)	75
2.5.1	Local SGLs.....	77
2.5.2	Extended SGLs (ESGLs)	77
2.6	Memory Address Space	78
2.6.1	MEMBASE-III Inbound Window Shifting	79
2.7	Host Buffer Alignment	80
3	Functional Operation.....	81
3.1	SAS Port Instantiation, Port Context and PORT_ID	81
3.2	Device Handle and DEVICE_ID.....	82
3.2.1	Initiator Mode DEVICE_ID	83
3.2.2	Target Mode DEVICE_ID	83
3.2.3	Combo Initiator and Target Mode DEVICE_ID.....	83
3.3	SAS Discovery	85
3.4	SAS Assists	86
3.5	SSP Initiator Write Operations	86
3.5.1	SSP Initiator Write Using an Extended Gather List.....	87
3.5.2	SSP Initiator Write Using a Local Gather List	88
3.6	SSP Initiator Read Operations	90
3.6.1	SSP Initiator Read Using an Extended Scatter List.....	90
3.6.2	SSP Initiator Read Using a Local Scatter List.....	92
3.7	SSP Target Write Operations	93
3.7.1	SSP Target Write Using an Extended Gather List	95
3.7.2	SSP Target Write Using a Local Gather List	97
3.8	SSP Target Read Operations	98
3.8.1	SSP Target Read Using an Extended Scatter List.....	98
3.8.2	SSP Target Read Using a Local Scatter List	101
3.9	SMP Request Operations	101
3.9.1	SMP Request Using Extended Request and Response Buffers	101
3.9.2	SMP Request Using Local (Embedded) Request and Response	104
3.10	SSP Initiator Task Management Operation	105
3.11	SSP Target Task Management Operation	106
3.12	SATA Assists.....	108
3.13	SATA Host Write Operations	108
3.13.1	SATA Host Write Using an Extended Gather List	110
3.13.2	SATA Host Write Using a Local Gather List	112

3.14	SATA Host Read Operations	113
3.14.1	SATA Host Read Using an Extended Scatter List	114
3.14.2	SATA Host Read Using a Local Scatter List	116
3.15	Firmware/Partition Update	117
3.15.1	Flash Image Verification	119
3.15.2	Firmware Update Logic	120
3.16	GPIO Operation	121
3.17	Event Log Operation	122
3.17.1	Event Log Header	122
3.17.2	Event Log Entry	123
3.18	High Priority Operation and Normal Priority IOMB Processing	124
3.18.1	Inbound Processing	125
3.18.2	Outbound Processing	126
3.19	Host Interrupt Usage Model	127
3.19.1	Setting Up Outbound Queue and Interrupt Vector Mapping	128
3.19.2	Legacy INT-X Usage Model	128
3.19.3	MSI or MSI-X 1-to-1 Mapping of Interrupt Vectors to OQs	128
3.19.4	MSI or MSI-X Interrupt Vector Shared by Multiple OQs	129
3.19.5	Interrupt Coalescing/Delay	129
3.20	Transport Layer Retry (TLR) Handling	130
3.20.1	SAS 1.1 Transport Layer Retry	130
3.20.2	SAS 2.0 Transport Layer Retry	131
3.21	Host Direct Access (HDA) Mode	131
3.21.1	Bringing the Boot ROM Into HDA Mode	132
3.21.2	Boot ROM HDA Protocol – Command and Response	133
3.21.3	HDA ILA Protocol – Command and Response/Status	137
3.21.4	Host and SPC 8x6G Initialization Sequence	138
3.22	SMP Request Serialization	142
3.23	Expansion ROM Support	145
3.24	Customizing Firmware Behavior	145
3.24.1	IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY	145
3.24.2	IOP_EVENT_IT_NEXUS_LOSS	146
3.24.3	IOP_EVENT_PHY_LOCKUP_DATAOUT	146
3.24.4	IOP_EVENT_PHY_DOC_ABORT_TMO	147
3.24.5	RETURN_ERROR_ON_OUT_OF_IOST_RESOURCE	147

3.24.6	ENABLE_OUTBOUND_PROCESSING_FAIRNESS	147
3.25	SGPIO Operation	148
4	Initialization and Configuration.....	149
4.1	Power-up and Initialization	151
4.2	SPC 8x6G Bootstrap Configuration	151
4.3	Configuration SEEPROM.....	152
4.3.1	General Configuration Table	152
4.4	Power-On Self-Test (POST).....	152
4.4.1	CPU Instruction and Data Caches	153
4.4.2	Randomly Accessible Memory Tests.....	153
4.5	Flash Memory Format	154
4.6	Flash Memory Partition Format.....	155
4.7	PCIe System Configuration.....	157
4.8	Firmware Image Load	157
4.9	PCI Configuration Initialization	158
4.10	SPC 8x6G Reset.....	158
4.11	SPC 8x6G Initialization	158
4.11.1	Verifying Host-SPC 8x6G Communication.....	159
4.12	PHY Initialization and Port Instantiation	160
4.12.1	PHY Initialization	160
4.12.2	Port Instantiation	161
4.13	Unloading the Host Driver	161
5	Message Passing Interface (MPI) Configuration	162
5.1	MPI Configuration Table Access	162
5.2	MPI Configuration Table	163
5.2.1	MPI Main Configuration Table Fields	164
5.2.2	MPI General Status Table Fields.....	181
5.2.3	MPI Inbound Queue Configuration Table Fields	188
5.2.4	MPI Outbound Queue Configuration Table Fields	191
5.2.5	MPI SAS PHY Analog Setup Table Fields	196
5.2.6	MPI State and Configuration Table Reading and Writing Sequence	198
6	Common IOMB Header	203
7	Inbound Messages.....	206
7.1	ECHO Command	207
7.2	PHY_START Command.....	208

7.3	PHY_STOP Command.....	210
7.4	SSP_INI_IO_START Command.....	211
7.5	SSP_INI_TM_START Command	215
7.6	SSP_INI_EXT_IO_START Command	218
7.7	DEVICE_HANDLE_ACCEPT Command.....	220
7.8	SSP_TGT_IO_START Command.....	222
7.9	SSP_TGT_RESPONSE_START Command.....	225
7.10	SSP_ABORT Command	228
7.11	DREGISTER_DEVICE_HANDLE Command.....	231
7.12	GET_DEVICE_HANDLE Command	232
7.13	SMP_REQUEST Command	234
7.14	SMP_ABORT Command.....	237
7.15	REGISTER_DEVICE Command.....	239
7.16	SATA_HOST_IO_START Command	242
7.17	SATA_ABORT Command	245
7.18	LOCAL_PHY_CONTROL Command.....	248
7.19	GET_DEVICE_INFO Command	250
7.20	FW_FLASH_UPDATE Command.....	251
7.21	GPIO Command.....	254
7.22	SAS_DIAG_MODE_START_END Command.....	257
7.23	SAS_DIAG_EXECUTE Command	258
7.24	SAS_HW_EVENT_ACK Command.....	267
7.25	GET_TIME_STAMP Command	270
7.26	PORT_CONTROL Command	271
7.27	GET_NVMD_DATA Command	274
7.27.1	Getting VPD Data.....	275
7.27.2	Getting Data from a TWI Device	275
7.27.3	Getting Register Dump Information from Flash Memory	275
7.27.4	Getting Event Log Information from Flash Memory	276
7.27.5	Reading Back The Expansion ROM Image from Flash Memory	276
7.28	SET_NVMD_DATA Command	279
7.29	SET_DEVICE_STATE Command	283
7.30	GET_DEVICE_STATE Command.....	285
7.31	SET_DEVICE_INFO Command.....	286
7.32	SAS_RE_INITIALIZATION Command	289

7.33	SGPIO_REGISTER Command.....	292
7.34	PCI_DIAG_EXECUTE Command.....	294
8	Outbound Messages	298
8.1	ECHO Response.....	299
8.2	SAS_HW_EVENT Notification	300
8.3	SSP_COMPLETION Response.....	310
8.4	SMP_COMPLETION Response	315
8.5	LOCAL_PHY_CONTROL Response	319
8.6	DEVICE_REGISTRATION Response.....	320
8.7	DREGISTER_DEVICE_HANDLE Response	323
8.8	GET_DEVICE_HANDLE Response	324
8.9	SATA_COMPLETION Response	325
8.10	SATA_EVENT Notification.....	329
8.11	SSP_EVENT Notification	334
8.12	DEVICE_HANDLE_ARRIVED Notification	338
8.13	SSP_REQUEST_RECEIVED Notification	340
8.14	GET_DEVICE_INFO Response.....	341
8.15	FW_FLASH_UPDATE Response	343
8.16	GPIO Response	345
8.17	GPIO_EVENT Notification	348
8.18	GENERAL_EVENT Notification	349
8.19	SSP_ABORT Response.....	350
8.20	SATA_ABORT Response.....	352
8.21	SAS_DIAG_MODE_START_END Response	353
8.22	SAS_DIAG_EXECUTE Response	354
8.23	GET_TIME_STAMP Response.....	356
8.24	SAS_HW_EVENT_ACK Response	357
8.25	PORT_CONTROL Response.....	358
8.26	SKIP_ENTRIES_EVENT Notification.....	359
8.27	SMP_ABORT Response	359
8.28	GET_NVMD_DATA Response	362
8.29	SET_NVMD_DATA Response	365
8.30	DEVICE_HANDLE_REMOVED Notification	368
8.31	SET_DEVICE_STATE Response	369
8.32	GET_DEVICE_STATE Response	370

8.33	SET_DEVICE_INFO Response	371
8.34	SAS_RE_INITIALIZATION Response.....	373
8.35	SGPIO Response.....	375
8.36	PCIE_DIAG_EXECUTE Response.....	377
9	PCIe Registers	379
9.1	Programming Interface.....	379
9.2	PCIe Configuration Space.....	379
9.2.1	PCIe Configuration Register Map	379
9.2.2	PCIe MSI-X Vector Table Address Map	381
9.2.3	Register Access and Reset Values	381
9.2.4	Vendor Identification Configuration Register.....	384
9.2.5	Device Identification Register.....	384
9.2.6	Configuration Command Register.....	385
9.2.7	Configuration Status Register	387
9.2.8	Revision Register	389
9.2.9	Class Code Register	390
9.2.10	Cache Line Size Register.....	390
9.2.11	Master Latency Timer Register	391
9.2.12	Header Type Register	391
9.2.13	Built-In Self Test Register	392
9.2.14	Memory Base Address I Register	393
9.2.15	Memory Base Address II Register	394
9.2.16	Memory Base Address III Register	395
9.2.17	Memory Base Address IV Register	396
9.2.18	Subsystem ID and Subsystem Vendor ID Register	397
9.2.19	ROM Base Address Register	397
9.2.20	Configuration Capabilities Pointer Register	398
9.2.21	Interrupt Line Register.....	398
9.2.22	Interrupt Pin Register	399
9.2.23	Minimum Grant Register	399
9.2.24	Maximum Latency Register.....	400
9.2.25	Power Management Capabilities Identifier Register	400
9.2.26	Power Management Next Capabilities Register	401
9.2.27	Power Management Capabilities Register.....	401
9.2.28	Power Management Control and Status Register.....	402

9.2.29	Message Signaled Interrupts Capabilities Identifier Register	404
9.2.30	Message Signaled Interrupts Next Capability Register.....	404
9.2.31	Message Control Register.....	405
9.2.32	Message Lower Address Register	406
9.2.33	Message Upper Address Register	406
9.2.34	Message Data Register.....	407
9.2.35	PCIe Capabilities Identifier Register	408
9.2.36	PCIe Next Capability Register.....	409
9.2.37	PCIe Capabilities Register	409
9.2.38	Device Capabilities Register	410
9.2.39	Device Control Register	412
9.2.40	Device Status Register.....	414
9.2.41	Link Capabilities Register.....	415
9.2.42	Link Control Register.....	416
9.2.43	Link Status Register	418
9.2.44	MSI-X Capabilities Register	419
9.2.45	Table Offset Register.....	420
9.2.46	PBA Offset Register	421
9.2.47	Enhanced Capability Header Register.....	422
9.2.48	Uncorrectable Error Status Register	423
9.2.49	Uncorrectable Error Mask Register.....	425
9.2.50	Uncorrectable Error Severity Register	426
9.2.51	Correctable Error Status Register	428
9.2.52	Correctable Error Mask Register.....	429
9.2.53	Advanced Error Capabilities and Control Register	430
9.2.54	Header Log Register	431
10	Chip Registers	432
10.1.1	Configuration Modes	432
10.2	Messaging Unit Registers	432
10.2.1	Inbound Doorbell Register	434
10.2.2	Inbound Doorbell Clear Register.....	435
10.2.3	Outbound Doorbell Register.....	436
10.2.4	Outbound Doorbell Clear Register	436
10.2.5	Scratchpad 0 Register.....	437
10.2.6	Scratchpad 1 Register.....	438

10.2.7	Scratchpad 2 Register.....	439
10.2.8	Scratchpad 3 Register.....	440
10.2.9	Host Scratchpad 0 Register	441
10.2.10	Host Scratchpad 1 Register	442
10.2.11	Host Scratchpad 2 Register	442
10.2.12	Host Scratchpad 3 Register	443
10.2.13	Host Scratchpad 4 Register	443
10.2.14	Host Scratchpad 5 Register	443
10.2.15	Host Scratchpad 6 Register	443
10.2.16	Host Scratchpad 7 Register	443
10.2.17	Outbound Doorbell Mask Register.....	444
10.3	SPC 8x6G Top-level Registers.....	444
10.3.1	Boot Strapping Bit Register.....	445
10.3.2	Device ID Register	445
10.3.3	Device Revision Register	445
10.3.4	SAS CSU Lock Detect Monitor Status Register.....	446
10.3.5	Device Test Register	447
10.3.6	Device LCLK Generation	448
10.3.7	SPC Reset Register	449
10.3.8	PCIe Event Interrupt Enable Register	451
10.3.9	PCIe Event Interrupt Register	453
10.3.10	PCIe Error Interrupt Enable Register	454
10.3.11	PCIe Error Interrupt Register	459
10.3.12	Inbound AXI Translation Lower Address – Window 2 Register.....	461
10.3.13	Inbound AXI Translation Upper Address – Window 2 Register.....	462
10.3.14	Message Unit Outbound Doorbell Auto Clear Register	462
10.3.15	Interrupt Coalescing Timer Register	463
10.3.16	Interrupt Coalescing Control Register.....	463
10.4	PCIe Gen2 PCS/PMA Module Registers	464
10.4.1	Test Control/Status Register.....	464
10.4.2	Lane N TRS Control Register	465
10.4.3	Lane N Transmitter Mode Register	467
10.4.4	Lane N Transmitter Control Register	470
10.4.5	Lane N Receiver RX Mode 1 Register	471
10.4.6	Lane N Receiver RX Mode 2 Register.....	472

10.4.7	Lane N Receiver Status Register	473
10.4.8	Lane N Error Counter Control Register.....	473
10.4.9	Lane N Disparity Error Count Register	474
10.4.10	Lane N Code Violation Error Count Register	475
10.4.11	Lane N PRBS and Test Pattern Error Count Register	476
10.4.12	Lane N Diagnostics Configuration Register	477
10.4.13	Lane N PRBS Initial Value Register	479
10.4.14	Lane N PRBS Error Insertion Register	480
10.4.15	Lane N Test Pattern Insertion Word 1_1 Register	480
10.4.16	Lane N Test Pattern Insertion Word 1_2 Register	481
10.4.17	Lane N Test Pattern Insertion Word 2_1 Register	482
10.4.18	Lane N Test Pattern Insertion Word 2_2 Register	483
10.5	Octal SAS/SATA Port Subsystem Registers	483
10.5.1	OSSP Control 1 Register	484
10.5.2	Timer Control 0 Register	486
10.5.3	Timer Control 2 Register	486
10.5.4	Transmitter Per Port Configuration 1 SAS_SATA G1 Register	487
10.5.5	Transmitter Per Port Configuration 1 SAS_SATA G2 Register	493
10.5.6	Transmitter Per Port Configuration 1 SAS_SATA G3 Register	499
10.5.7	Receiver Per Port Configuration 1 SAS_SATA G1G2 Register	505
10.5.8	Receiver Per Port Configuration 1 SAS_SATA G3 Register	508
10.5.9	Global Configuration 1 Register	511
10.5.10	Transmitter Configuration 1 Register	512
10.5.11	Receiver Configuration 1 Register	517
10.5.12	Receiver Configuration 2 Register	518
10.6	SAS/SATA PHY Layer Registers.....	519
10.6.1	Connection Status Register.....	521
10.6.2	Error Interval Thresholds Register	523
10.6.3	Interrupt Status 1 Register	524
10.6.4	Interrupt Values 1 Register.....	528
10.6.5	Counter Configuration Register	529
10.6.6	Invalid DWord Count Register.....	530
10.6.7	Disparity Error Count Register	531
10.6.8	Code Violation Error Count Register.....	532
10.6.9	Loss of DWord Synchronization Count Register.....	533

10.6.10	PHY Reset Failed Count Register.....	534
10.6.11	PRBS and Test Pattern Error Count Register	535
10.6.12	SAS2 RX Error Count Register.....	536
10.6.13	Diagnostics Configuration Register.....	536
10.6.14	Error Insertion Register 1	539
10.6.15	Error Insertion Register 2.....	539
10.6.16	Test Pattern Insertion Word 1_1 Register	540
10.6.17	Test Pattern Insertion Word 1_2 Register	541
10.6.18	Test Pattern Insertion Word 2_1 Register	542
10.6.19	Test Pattern Insertion Word 2_2 Register	543
10.6.20	SAS 2 Settings (Local) Register.....	544
10.6.21	Maximum AIP Allowed Register	544
10.6.22	Open Retry Interval Register.....	545
10.7	SAS/SATA Port Adapter Link Registers	546
10.7.1	Control Register	547
10.7.2	Performance Counter 1 Control Register.....	549
10.7.3	Performance Counter 1 Threshold Register	552
10.7.4	Performance Counter 1 Count Register.....	552
10.7.5	Performance Counter 2 Control Register.....	552
10.7.6	Performance Counter 2 Threshold Register	553
10.7.7	Performance Counter 2 Count Register.....	554
10.7.8	Peak Detector Control Register	554
10.7.9	Peak Detector 1 Count Register	555
10.7.10	Peak Detector 1 Threshold Register	556
10.7.11	Timer Enables Register.....	556
10.7.12	Rate Control Register.....	557
10.7.13	Connection Time Register.....	558
10.7.14	General Purpose Register.....	559
10.8	SPC 8x6G Global Shared Memory Registers	560
10.8.1	GSM Configuration and Reset Register.....	561
10.8.2	GSM RAM ECC Double Bit Error Indication Register	563
10.8.3	GSM Read Address Parity Check Enable Register	565
10.8.4	GSM Write Address Parity Check Enable Register	565
10.8.5	GSM Write Data Parity Check Enable Register	566
10.8.6	GSM Read Address Parity Error Indication Register	567

10.8.7	GSM Write Address Parity Error Indication Register	567
10.8.8	GSM Write Data Parity Error Indication Register.....	568
10.8.9	RB6 Access Register	568
10.9	MBIC Registers	569
10.9.1	NMI Enable VPE0 AAP1 Register.....	569
10.9.2	NMI Enable VPE0 IOP Register.....	571
10.10	Miscellaneous Registers.....	571
	10.10.1 GPIO-0 Output Control Register	572
11	Error Information	573
11.1	PCIe Errors.....	573
11.1.1	Unsupported Requests	573
11.2	Device Specific Fatal Errors	574
11.2.1	Block Direct Memory Access (BDMA) Fatal Errors.....	578
11.2.2	Global Shared Memory (GSM) Fatal Errors.....	580
11.2.3	MIPS34K Bridge and Interrupt Controller (MBIC) Fatal Errors	583
11.2.4	PMIC1 Error (AXI Master Slave)	587
11.2.5	PMIC2 Error (ECC RAM)	589
11.2.6	PMIC Event Error	590
11.2.7	Octal SAS/SATA Port (OSSP) Fatal Errors	591
11.2.8	Hardened SAS/SATA Transport (HSST) Fatal Errors	593
11.2.9	Processor Complex System (PCS) Fatal Errors	594
11.2.10	PCIe General Fatal Error	594
11.2.11	ILA Image Loading Error	595
11.2.12	Firmware Assert Fatal Error	596
11.2.13	Firmware Watchdog Timer Fatal Error	597
11.2.14	Firmware General Fatal Errors.....	598
11.2.15	Firmware Fatal Errors Register Dump	599
11.3	Device Specific Recoverable/Correctable Errors	612
11.4	Device Specific Fatal Error Recovery Procedures	612
11.4.1	Chip Status Determination Before Soft-Reset.....	613
11.4.2	Soft Reset Recovery (Normal Mode)	613
11.4.3	Soft Reset Recovery (HDA Mode)	619
11.4.4	Chip Reset.....	623
11.5	SAS/SATA Error Recovery Procedures.....	623
	11.5.1 PHY Down Handler (External Trigger)	623

11.5.2	Local PHY Control Link/Hard Reset Handler (Host Initiated)	629
11.5.3	BROADCAST CHANGE Handler.....	640
11.5.4	PHY Error Handler	641
11.5.5	Hard Reset Received Handler	643
11.5.6	Device States Related Error Handler	647
11.6	Detailed Descriptions of the SAS Error Conditions.....	655
11.6.1	SAS Physical Layer.....	655
11.6.2	SAS Link Layer	657
11.6.3	SAS Port Layer	667
11.6.4	SSP Transport Layer (Initiator Port)	668
11.6.5	SSP Transport Layer (Target Port).....	670
11.6.6	SMP Transport Layer	672
11.7	Detailed Description of SATA Error Conditions	672
11.7.1	SATA Physical Layer	673
11.7.2	SATA Link Layer	673
11.7.3	SATA Transport Layer	676
Glossary		678

List of Figures

Figure 1	Fabric-Attached Storage System Application	64
Figure 2	PM8001 SPC 8x6G Block Diagram	65
Figure 3	OSSP Internal Block Diagram.....	67
Figure 4	Circular Queue	71
Figure 5	OQ Full and Empty Example.....	72
Figure 6	IQ With Five Entries Produced.....	73
Figure 7	OQ With Five Entries Produced	75
Figure 8	Extended Scatter/Gather List (ESGL).....	77
Figure 9	SSP Initiator Write Operations Flow Diagram.....	86
Figure 10	SSP Initiator Write Example - Extended Gather List.....	87
Figure 11	SSP Initiator Write Example - Local Buffer Descriptor.....	89
Figure 12	SSP Initiator Read Operations Flow Diagram.....	90
Figure 13	SSP Initiator Read Example - Extended Scatter List	91
Figure 14	SSP Initiator Read Example - Local Buffer Descriptor	93
Figure 15	SSP Target Write Operations Flow Diagram	93
Figure 16	SSP Target Write Example - Extended Gather List	95
Figure 17	SSP Target Write Example - Local Gather List.....	97
Figure 18	SSP Target Read Operation Flow Diagram.....	98
Figure 19	SSP Target Read Example - Extended Scatter List	99
Figure 20	SSP Target Read Example - Local Scatter List.....	101
Figure 21	SMP Request Example - Extended Request and Response Byte Buffers.....	102
Figure 22	SMP Initiator with Extended Request and Response Buffers Flow Diagram.....	103
Figure 23	SMP Request Example – with Local Request and Response Bytes	104
Figure 24	SMP Initiator with Local Request and Response Bytes Flow Diagram.....	105
Figure 25	SSP Initiator Task Management Flow Diagram	106
Figure 26	SSP Target Task Management Operation Flow Diagram	107
Figure 27	Host Write Operation Flow Diagram (PIO Data Transfer Mode)	108
Figure 28	SATA Host Write Operation Flow Diagram (DMA Data Transfer Mode)	109
Figure 29	SATA Host Write Operation Flow Diagram (FPDMA Data Transfer Mode)	109

Figure 30	SATA Host Write Example - Extended Gather List.....	110
Figure 31	SATA Host Write Example - Local Gather List	112
Figure 32	Host Read Operation Flow Diagram (PIO Data Transfer Mode)	113
Figure 33	SATA Host Write Operation Flow Diagram (DMA Data Transfer Mode)	113
Figure 34	SATA Host Read Operation Flow Diagram (FPDMA Data Transfer Mode)	114
Figure 35	SATA Host Read Example - Extended Gather List.....	114
Figure 36	SATA Host Read Example - Local Gather List	116
Figure 37	Partition Binary Files – Frame Flow and Image Length Value.....	119
Figure 38	Event Log Buffer Format.....	122
Figure 39	Messaging Unit Queue Processing.....	124
Figure 40	Boot ROM HDA State Diagram.....	136
Figure 41	SMP Queuing and Serialization Diagram	144
Figure 42	SPC 8x6G Initialization Flow.....	150
Figure 44	Flash Memory Map	154
Figure 45	SPC 8x6G Firmware Image Partition Format	156
Figure 46	Firmware Image Flag Byte Fields	157
Figure 47	PHY Initialization Flow	160
Figure 48	IOMB Command Header.....	203
Figure 49	Line-side and System-side Loopbacks	265
Figure 50	Sample Register Dump Snapshot.....	599
Figure 51	PHY Down (of the Last PHY in a Port) That Recovered.....	625
Figure 52	PHY Down Sequence with No Recovery After the Port Recovery Time	626
Figure 53	Last PHY Down in a Port with Port Recovery Time Set to Zero	627
Figure 54	PHY Down Sequence of One of the PHYs in a Wide Port	628
Figure 55	Narrow Port Successful Local PHY Control Link/Hard Reset.....	630
Figure 56	Narrow Port Unsuccessful Local PHY Control Link/Hard Reset.....	631
Figure 57	Wide Port Successful Local PHY Control Link Reset	632
Figure 58	Wide Port Partial Successful Local PHY Control Link Reset	634
Figure 59	Wide Port Unsuccessful Local PHY Control Link Reset	636
Figure 60	Wide Port Successful Local PHY Control Hard Reset Without Port Recovery Timer.....	638
Figure 61	Wide Port Successful Local PHY Control Hard Reset With Port Recovery Timer.....	639

Figure 62	BROADCAST CHANGE Acknowledgement Sequence	641
Figure 63	PHY Error Acknowledgement Sequence	642
Figure 64	Narrow Port– Hard Reset Received – Port Recovery Time Enabled	644
Figure 65	Narrow Hard Reset Received - Port Recovery Time Disabled	645
Figure 66	Wide Port – Hard Reset Received – Port Recovery Time Enabled/Disabled.....	646
Figure 67	Recovery with Device Reset and SET_DEVICE_STATE Command	648
Figure 68	Task Management Recovery with DS=1 and Normal Priority IQ.....	649
Figure 69	Task Management Recovery with DS=1 and High Priority IQ	650
Figure 70	Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices	651
Figure 71	Head-Of-Line Blocking for SATA Protocol (NCQ Command).....	653
Figure 72	Head-Of-Line Blocking for SATA Protocol (NCQ Data)	654

List of Tables

Table 1	Numeric Notation.....	60
Table 2	Big Endian	69
Table 3	Little Endian.....	69
Table 4	Scatter/Gather List Element (L/A Pair) Data Structure	76
Table 5	SGL Element (L/A Pair) Fields	76
Table 6	Default PCIe BAR Configuration per Detected Device Type	78
Table 7	Alignment Requirements	80
Table 8	Device State Definition	82
Table 9	IOMB Commands for SSP Initiator I/O Operations	87
Table 10	IOMB Commands for SSP Target I/O Operations	95
Table 11	SATA Read/Write Operation Modes	111
Table 12	Firmware/Partition Image Header	117
Table 13	GPIO Operation Precedence	121
Table 14	GPIO LED Control.....	121
Table 15	Event Log Header Format.....	123
Table 16	Event Log Entry Format	123
Table 17	Host Controlled Interrupt Registers	127
Table 18	SAS-2 TLR Mode Encoding	131
Table 19	SPC 8x6G Processing of TLR for SSP Command.....	131
Table 20	Boot ROM HDA Protocol Command Format.....	133
Table 21	Boot ROM HDA Protocol Response Format	134
Table 22	Boot ROM HDA Commands.....	135
Table 23	Boot ROM HDA Responses	135
Table 24	HDA-ILA Commands in the Host Scratchpad 3 Register.....	137
Table 25	HDA-ILA Responses or State in the Scratchpad 0 Register.....	137
Table 26	Host and SPC 8x6G Initialization Sequence	138
Table 27	Bootstrap Configuration.....	151
Table 28	General Configuration	152
Table 35	Flash Partitions.....	154
Table 36	Scratchpad 1 Register Status Indication for AAP1	162
Table 37	Scratchpad 2 Register Status Indication for IOP.....	163
Table 38	MPI Configuration Table – Main Part	164
Table 39	General Status Table (GST).....	182

Table 40	Inbound Queue Configuration Table (IQCT)	188
Table 41	Outbound Queue Configuration Table (OQCT)	191
Table 42	SAS PHY Analog Setup Table (SPAST).....	196
Table 43	IOMB Command Header Fields	203
Table 44	Summary of Inbound Operation Codes.....	206
Table 45	ECHO Command Format.....	207
Table 46	ECHO Command Fields.....	207
Table 47	PHY_START Command Format	209
Table 48	PHY_START Command Fields.....	209
Table 49	PHY_STOP Command Format	211
Table 50	PHY_STOP Command Fields	211
Table 51	SSP_INI_IO_START Command Format.....	212
Table 52	SSP_INI_IO_START Command Fields.....	212
Table 53	SSP_INI_TM_START Command Format	215
Table 54	SSP_INI_TM_START Command Fields	216
Table 55	SSP_INI_EXT_IO_START Command Format.....	218
Table 56	SSP_INI_EXT_IO_START Command Fields.....	218
Table 57	DEVICE_HANDLE_ACCEPT Command Format.....	221
Table 58	DEVICE_HANDLE_ACCEPT Command Fields	221
Table 59	SSP_TGT_IO_START Command Format	223
Table 60	SSP_TGT_IO_START Command Fields	223
Table 61	SSP_TGT_RESPONSE_START Command Format.....	226
Table 62	SSP_TGT_RESPONSE_START Command Fields.....	226
Table 63	SSP_ABORT Scope of Abort (SCP)	229
Table 64	SSP_ABORT Command Format.....	230
Table 65	SSP_ABORT Command Fields	230
Table 66	DREGISTER_DEVICE_HANDLE Command Format.....	232
Table 67	DREGISTER_DEVICE_HANDLE Command Fields	232
Table 68	GET_DEVICE_HANDLE Command Format.....	233
Table 69	GET_DEVICE_HANDLE Command Fields	233
Table 70	SMP_REQUEST Command Format	234
Table 71	SMP_REQUEST Command Fields	234
Table 72	SMP_ABORT Scope of Abort (SCP)	237
Table 73	SMP_ABORT Command Format	238
Table 74	SMP_ABORT Command Fields	238

Table 75	REGISTER_DEVICE Command Format	240
Table 76	REGISTER_DEVICE Command Fields	240
Table 77	SATA_HOST_IO_START Command Format	243
Table 78	SATA_HOST_IO_START Command Fields	243
Table 79	SATA_ABORT Scope of Abort (SCP)	246
Table 80	SATA_ABORT Command Format	247
Table 81	SATA_ABORT Command Fields	247
Table 82	LOCAL_PHY_CONTROL Command Format	248
Table 83	LOCAL_PHY_CONTROL Command Fields	249
Table 84	GET_DEVICE_INFO Command Format	250
Table 85	GET_DEVICE_INFO Command Fields	250
Table 86	FW_FLASH_UPDATE Command Format	251
Table 87	FW_FLASH_UPDATE Command Fields	252
Table 88	GPIO Command Format	254
Table 89	GPIO Command Fields	254
Table 90	SAS_DIAG_MODE_START_END Command Format	257
Table 91	SAS_DIAG_MODE_START_END Command Fields	258
Table 92	SAS_DIAG_EXECUTE Command Format	259
Table 93	SAS_DIAG_EXECUTE Command Fields	259
Table 94	SAS Diagnostic Command Type and Command Description Valid Combination	262
Table 95	Command Descriptors for Command Types DIAG_OPRN_PERFORM, DIAG_OPRN_STOP, THRESHOLD_SPECIFY and RECEIVE_ENABLE	263
Table 96	Command Descriptors for Command Types DIAG_REPORT_GET and ERR_CNT_RESET	264
Table 97	SAS Diagnostic Command — Required Argument	264
Table 98	SAS_HW_EVENT_ACK Command Format	268
Table 99	SAS_HW_EVENT_ACK Command Fields	268
Table 100	GET_TIME_STAMP Command Format	271
Table 101	GET_TIME_STAMP Command Field	271
Table 102	PORT_CONTROL Command Format	272
Table 103	PORT_CONTROL Command Fields	272
Table 104	GET_NVMD Command Format	277
Table 105	GET_NVMD Command Fields	277
Table 106	SET_NVMD Command Format	280

Table 107 SET_NVMD Command Fields.....	281
Table 108 SET_DEVICE_STATE Command Format	284
Table 109 SET_DEVICE_STATE Command Fields	284
Table 110 GET_DEVICE_STATE Command Format.....	285
Table 111 GET_DEVICE_STATE Command Fields.....	286
Table 112 SET_DEVICE_INFO Command Format	286
Table 113 SET_DEVICE_INFO Command Fields	287
Table 114 SAS_RE_INITIALIZATION Command Format.....	289
Table 115 SAS_RE_INITIALIZATION Command Fields	290
Table 116 GPIO_REGISTER Command Format	292
Table 117 GPIO_REGISTER Command Fields	293
Table 118 PCI_DIAG_EXECUTE Command Format	294
Table 119 PCI_DIAG_EXECUTE Command Fields	295
Table 120 PCIe Diagnostic Command Type and Command Description Valid Combination	296
Table 121 Outbound Operation Codes	298
Table 122 ECHO Response Format	299
Table 123 ECHO Response Fields	299
Table 124 SAS_HW Event Format.....	301
Table 125 SAS_HW Event Fields	301
Table 126 SSP_COMPLETION Response Format.....	310
Table 127 SSP_COMPLETION Response Fields	311
Table 128 SMP_COMPLETION Response Format.....	316
Table 129 SMP_COMPLETION Response Fields	316
Table 130 LOCAL_PHY_CONTROL Response Format.....	319
Table 131 LOCAL_PHY_CONTROL Response Fields.....	320
Table 132 DEVICE_REGISTRATION Response Format	321
Table 133 DEVICE_REGISTRATION Response Fields	321
Table 134 DREGISTER_DEVICE_HANDLE Response Format	323
Table 135 DREGISTER_DEVICE_HANDLE Response Fields	323
Table 136 GET_DEVICE_HANDLE Response Format	324
Table 137 GET_DEVICE_HANDLE Response Fields	324
Table 138 SATA_COMPLETION Response Format	325
Table 139 SATA_COMPLETION Response Fields	325
Table 140 SATA Event Format	330

Table 141	SATA Event Fields	330
Table 142	SSP Event Format.....	335
Table 143	SSP Event Fields	335
Table 144	DEVICE_HANDLE_ARRIVED Event Format.....	339
Table 145	DEVICE_HANDLE_ARRIVED Event Fields	339
Table 146	SSP_REQUEST_RECEIVED Event Format.....	340
Table 147	SSP_REQUEST_RECEIVED Event Fields	340
Table 148	GET_DEVICE_INFO Response Format	342
Table 149	GET_DEVICE_INFO Response Fields	342
Table 150	FW_FLASH_UPDATE Response Format.....	344
Table 151	FW_FLASH_UPDATE Response Fields.....	344
Table 152	GPIO Response Format.....	346
Table 153	GPIO Response Fields.....	346
Table 154	GPIO Response Format.....	348
Table 155	GPIO Response Fields.....	348
Table 156	GENERAL_EVENT Notification Format.....	349
Table 157	GENERAL_EVENT Notification Fields.....	349
Table 158	SSP_ABORT Response Format	350
Table 159	SSP_ABORT Response Fields	351
Table 160	SATA_ABORT Response Format	352
Table 161	SATA_ABORT Response Fields	352
Table 162	SAS_DIAG_MODE_START_END Response Format	354
Table 163	SAS_DIAG_MODE_START_END Response Fields	354
Table 164	SAS_DIAG_EXECUTE Response Format.....	355
Table 165	SAS_DIAG_EXECUTE Response Fields	355
Table 166	GET_TIME_STAMP Response Format	356
Table 167	GET_TIME_STAMP Response Fields	356
Table 168	SAS_HW_EVENT_ACK Response Format	357
Table 169	SAS_HW_EVENT_ACK Response Fields	357
Table 170	PORT_CONTROL Response Format	358
Table 171	PORT_CONTROL Response Fields	358
Table 172	SKIP_ENTRIES_EVENT Notification.....	359
Table 173	SKIP_ENTRIES_EVENT Notification Field.....	359
Table 174	SMP_ABORT Response Format.....	360
Table 175	SMP_ABORT Response Fields	361

Table 176	GET_NVMD_DATA Response Format	362
Table 177	GET_NVMD_DATA Response Fields	362
Table 178	SET_NVMD Response Format	365
Table 179	SET_NVMD Response Fields	366
Table 180	DEVICE_HANDLE_Removed Notification Format.....	368
Table 181	DEVICE_HANDLE_Removed Notification Fields	368
Table 182	SET_DEVICE_STATE Response Format.....	369
Table 183	SET_DEVICE_STATE Response Fields	369
Table 184	GET_DEVICE_STATE Response Format	370
Table 185	GET_DEVICE_STATE Response Fields	371
Table 186	SET_DEVICE_INFO Response Format.....	372
Table 187	SET_DEVICE_INFO Response Fields	372
Table 188	SAS_RE_INITIALIZATION Response Format.....	373
Table 189	SAS_RE_INITIALIZATION Response Fields.....	374
Table 190	SGPIO Response Format	375
Table 191	SGPIO Response Fields	376
Table 192	PCIE_DIAG_EXECUTE Response Format	377
Table 193	PCIE_DIAG_EXECUTE Response Fields	377
Table 194	PCIe Configuration Register Map	379
Table 195	PCIe MSI-X Vector Table Address Map.....	381
Table 196	Register Access and Reset Values	381
Table 197	Configuration Command Register	385
Table 198	Configuration Status Register	387
Table 199	Revision Register	389
Table 200	Header Type Register	392
Table 201	Power Management Capabilities Register Bits	402
Table 202	Power Management Control and Status Register Bits.....	403
Table 203	Message Control Register Bits.....	405
Table 204	Message Address Bits.....	406
Table 205	Message Data Register	407
Table 206	PCIe Capabilities Register Bits	409
Table 207	Device Capabilities Register Bits	410
Table 208	Device Control Register Bits	412
Table 209	Device Status Register Bits.....	414
Table 210	Link Capabilities Register Bits.....	415

Table 211	Link Control Register Bits	417
Table 212	Link Status Register Bits	418
Table 213	MSI-X Capabilities Register Bits	419
Table 214	Offset Register Bits	420
Table 215	PBA Offset Register Bits	421
Table 216	Enhanced Capability Header Register Bits	422
Table 217	Uncorrectable Error Status Register Bits	423
Table 218	Uncorrectable Error Mask Register Bits	425
Table 219	Uncorrectable Error Severity Register Bits	427
Table 220	Correctable Error Status Register Bits	428
Table 221	Correctable Error Mask Register Bits	429
Table 222	Advanced Error Capabilities and Control Register Bits	430
Table 223	Header Log Register Bits	431
Table 224	Chip Register Configuration Modes	432
Table 225	Messaging Unit Address Map	432
Table 226	Inbound Doorbell Register.....	434
Table 227	Inbound Doorbell Clear Register.....	435
Table 228	MSGU Scratchpad 1 Register	438
Table 229	MSGU Scratchpad 2 Register	439
Table 230	Scratchpad 3 Register when IOP_STATE is in Ready State.....	440
Table 231	SPC 8x6G Top-Level Address Map	444
Table 232	Boot Strapping Bit Register Bits	445
Table 233	Device ID Register Bits	445
Table 234	Device Revision Register Bits	446
Table 235	SAS CSU Lock Detect Monitor Interrupt Status Register Bits	446
Table 236	Device Test Register Bits	447
Table 237	Device Test Register Bits	448
Table 238	SPC Reset Register Bits	449
Table 239	PCIe Event Interrupt Enable Register Bits	451
Table 240	PCIe Event Interrupt Register Bits	453
Table 241	PCIe Error Interrupt Enable Register Bits	455
Table 242	PCIe Error Interrupt Register Bits	459
Table 243	Inbound AXI Translation Lower Address – Window 2 Register Bits	462
Table 244	Inbound AXI Translation Lower Address – Window 2 Register Bits	462
Table 245	MSGU Outbound Doorbell Auto Clear Register Bits.....	463

Table 246	Interrupt Coalescing Timer Register Bits	463
Table 247	Interrupt Coalescing Control Register Bits	464
Table 248	PCIe Gen2 PCS/PMA Module Address Map	464
Table 249	Test Control/Status Register Bits	465
Table 250	Lane 0 TRS Control Register Bits	466
Table 251	Lane 0 Transmitter Mode Register Bits.....	467
Table 252	Lane 0 Transmitter Control Register Bits	470
Table 253	Lane 0 Receiver RX Mode 1 Register Bits.....	471
Table 254	Lane 0 Receiver RX Mode 2 Register Bits.....	472
Table 255	Lane 0 Receiver Status Register Bits	473
Table 256	Lane 0 Error Counter Control Register Bits	473
Table 257	Lane 0 Disparity Error Count Register Bits	474
Table 258	Lane 0 Code Violation Error Count Register Bits.....	475
Table 259	Lane 0 PRBS and Test Pattern Error Count Register Bits.....	476
Table 260	Lane 0 Diagnostics Configuration Register Bits.....	477
Table 261	Lane 0 PRBS Initial Value Register Bits	479
Table 262	Lane 0 PRBS Error Insertion Register Bits	480
Table 263	Lane 0 Test Pattern Insertion Word 1_1 Register Bits.....	480
Table 264	Lane 0 Test Pattern Insertion Word 1_2 Register Bits.....	481
Table 265	Lane 0 Test Pattern Insertion Word 2_1 Register Bits.....	482
Table 266	Lane 0 Test Pattern Insertion Word 2_2 Register Bits.....	483
Table 267	Octal SAS/SATA Port Subsystem Address Map.....	483
Table 268	OSSP Control 1 Register Bits	484
Table 269	Timer Control 0 Register Bits	486
Table 270	Timer Control 2 Register Bits	486
Table 271	Transmitter Per Port Configuration 1 SAS_SATA G1 Register Bits	487
Table 272	Transmitter Per Port Configuration 1 SAS_SATA G2 Register Bits	493
Table 273	Transmitter Per Port Configuration 1 SAS_SATA G3 Register Bits	499
Table 274	Receiver Per Port Configuration 1 SAS_SATA G1G2 Register Bits.....	505
Table 275	Receiver Per Port Configuration 1 SAS_SATA G3 Register Bits	508
Table 276	Global Configuration 1 Register Bits	511
Table 277	Transmitter Configuration 1 Register Bits	512
Table 278	Receiver Configuration 1 Register Bits	517
Table 279	Receiver Configuration 2 Register Bits	518
Table 280	SAS/SATA PHY Layer Address Map	519

Table 281 Connection Status Register Bits	521
Table 282 Error Interval Thresholds Register Bits	523
Table 283 Interrupt Status 1 Register Bits	524
Table 284 Interrupt Values 1 Register Bits	528
Table 285 Invalid DWord Count Register Bits.....	529
Table 286 Invalid DWord Count Register Bits.....	530
Table 287 Disparity Error Count Register Bits	531
Table 288 Code Violation Error Count Register Bits.....	532
Table 289 Loss of DWord Synchronization Count Register Bits.....	533
Table 290 PHY Reset Failed Count Register Bits.....	534
Table 291 PRBS and Test Pattern Error Count Register Bits.....	535
Table 292 SAS2 RX Error Count Register Bits	536
Table 293 Diagnostics Configuration Register Bits.....	536
Table 294 Error Insertion Register 1 Register Bits.....	539
Table 295 Error Insertion Register 2 Register Bits.....	540
Table 296 Test Pattern Insertion Word 1_1 Register Bits.....	540
Table 297 Test Pattern Insertion Word 1_2 Register Bits.....	541
Table 298 Test Pattern Insertion Word 2_1 Register Bits.....	542
Table 299 Test Pattern Insertion Word 2_2 Register Bits.....	543
Table 300 SAS 2 Settings (Local) Register Bits.....	544
Table 301 Maximum AIP Allowed Bits	545
Table 302 Open Retry Interval Bits	545
Table 303 SAS/SATA Port Adapter Link Address Map.....	546
Table 304 Control Register Bits	547
Table 305 Performance Counter 1 Control Register Bits.....	550
Table 306 Performance Counter 1 Threshold Register Bits	552
Table 307 Performance Counter 1 Count Register Bits.....	552
Table 308 Performance Counter 2 Control Register Bits.....	553
Table 309 Performance Counter 2 Threshold Register Bits	553
Table 310 Perf 2 Count Register Bits.....	554
Table 311 Peak Detector Control Register Bits	554
Table 312 Peak Detector 1 Count Register Bits	555
Table 313 Peak Detector 1 Threshold Register Bits	556
Table 314 Timer Enables Register Bits.....	556
Table 315 Rate Control Register Bits.....	557

Table 316	Connection Time Register Bits	558
Table 317	General Purpose Register Bits	559
Table 318	Global Shared Memory Address Map	560
Table 319	GSM Configuration and Reset Register Bits	561
Table 320	RAM ECC Double Bit Error Indication Register Bits	563
Table 321	GSM Read Address Parity Check Enable Register Bits	565
Table 322	GSM Write Address Parity Check Enable Register Bits	566
Table 323	GSM Write Data Parity Check Enable Register Bits	566
Table 324	Read Address Parity Error Indication Register Bits	567
Table 325	Write Address Parity Error Indication Register Bits	567
Table 326	Write Data Parity Error Indication Register Bits	568
Table 327	RB6 Access Register Bits	568
Table 328	MBIC Address Map	569
Table 329	NMI Enable VPE0 AAP1 Register Bits	569
Table 330	MBIC EXT_NMI[31:0] Sources	570
Table 331	NMI Enable VPE0 IOP Register Bits	571
Table 332	Miscellaneous Registers Address Map	571
Table 333	GPIO-0 Output Control Register Bits	572
Table 334	Scratchpad 1 Register – AAP1/MSGU Status Indication for Fatal Error Case	575
Table 335	Scratchpad 2 Register – IOP Status Indication for Fatal Error Case	576
Table 336	BDMA Error Types	578
Table 337	BDMA Fatal Error Details – INTERRUPT 1 EVENT	579
Table 338	BDMA Fatal Error Details - INTERRUPT 2 EVENT	579
Table 339	BDMA Fatal Error Details - INTERRUPT 3 EVENT	580
Table 340	GSM_ERR Types	581
Table 341	GSM – DOUBLE_BIT_ECC_ERR Details	581
Table 342	GSM – READ_ADDRESS_PARITY_ERR Details	581
Table 343	GSM – WRITE_ADDRESS_PARITY_ERR Details	582
Table 344	GSM - WRITE_DATA_PARITY_ERR Details	582
Table 345	GSM – WSTRB_ERR Details	582
Table 346	MBIC_ERR Types	583
Table 347	MBIC - GSM_AXI_ERR Details	584
Table 348	MBIC - MXCBI_AXI_ERR Details	584
Table 349	MBIC - PCIE_AXI_ERR Details	584

Table 350 MBIC - INTERNAL_SET0_ERR Details.....	585
Table 351 MBIC - INTERNAL_SET1_ERR Details.....	586
Table 352 PMIC1_ERR – AXI_MASTER_SLAVE_ERR Details	587
Table 353 PMIC2_ERR – ECC_RAM_ERR Details	589
Table 354 PMIC_EVENT_ERROR Details	590
Table 355 OSSP_ERR Error Types	591
Table 356 OSSP - GSM_DECODE_RESPONSE_PARITY_ERR Details.....	591
Table 357 OSSP - FIFO_ECC_ERR.....	592
Table 358 OSSP - HSST_GLOBAL_ECC_ERR.....	592
Table 359 HSST_ERR Error Types	593
Table 360 HSST – TRANSPORT_ECC_ERR_E1_P[N] Details.....	593
Table 361 HSST – BDMA_TIMER_EXPIRED Details	594
Table 362 PCS_ERR Details	594
Table 363 PCIE_ERR – PCIE - Uncorrectable Error Status	595
Table 364 ILA Image Loading Error Details	595
Table 365 Firmware Assert Error Details	597
Table 366 Firmware Watchdog Timer Error Details.....	597
Table 367 Internal Registers Accessed For a Register Dump.....	604
Table 368 Soft Reset Sequence (Normal Mode)	614
Table 369 Soft Reset Sequence (HDA Mode)	619
Table 370 Abandon Class OPEN_REJECT Primitives	659
Table 371 Retry Class OPEN_REJECT Primitives	661
Table 372 SSP Initiator Invalid XFER_RDY Received.....	668
Table 373 SSP Initiator Invalid DATA Frame Received.....	669
Table 374 SSP Target Invalid DATA Frame Received	672

Preface

This manual is written for technical managers and engineers who are evaluating the PM8001 Tachyon SPC 8x6G controller or who are designing it into their SAS/SATA products. Readers wanting to understand the device functionality without delving into implementation specifics should concentrate on Sections 1 and 2.

Numeric Notation

This document uses the following numeric notation:

Table 1 Numeric Notation

Type	Notation	Example
Binary	'b' suffix	00100110b
Decimal	(default)	13
Hexadecimal	'0x' prefix 'x'hxx' prefix	0x5F 13'h0000

Use of the PM8001 SPC 8x6G

The PM8001 SPC 8x6G controller must be programmed and operated as defined in this manual. Otherwise, the device may enter an indeterminate state and/or produce unpredictable results.

In addition, if any of the SAS or SATA specifications are violated, the host is responsible for all error recovery.

Related Standards and Publications

Standards

1. American National Standard, Information Technology, *SCSI Block Commands – 2 (SBC-2)*. Project T10/1417-D, revision 16, November 13, 2004.
2. ANSI INCITS. T10/1601-D Revision 95 – *Serial Attached SCSI (SAS)*, May 30, 2005.
3. ANSI INCITS. T13/1532-D Volume 3 Revision 3 – *AT Attachment with Packet Interface – 7*, Volume 3 (ATA/ATAPI-7 V3), June 25, 2003.
4. *PCIe 2.0 Base Specification*, Revision 0.7, PCI-SIG, November, 2005.
5. Serial ATA Workgroup. *Serial ATA: High Speed Serialized Attachment*, Revision 1.0a - January 7, 2003.
6. Serial ATA Workgroup. *Serial ATA II: Extensions to Serial ATA*, Revision 1.0a - August 27, 2004.

7. Serial ATA Workgroup. *Serial ATA: Technical Errata 056 to Serial ATA*, Revision 1.0a - June 16, 2004.
8. Serial ATA Workgroup. *Serial ATA: Technical Errata 063 to Serial ATA*, Revision 1.0a - March 3, 2005.
9. SFF Committee. *SFF 8485 Specification for Serial GPIO(SGPIO) Bus, Revision 0.7* February 1, 2006.

PM8001 SPC 8x6G Documents

1. PMC-2080084, *PM8001 Tachyon SPC 8x6G 8-Port SAS/SATA Protocol Controller Product Brief*, issue 4.
2. PMC-2080082, *PM8001 Tachyon SPC 8x6G 8-Port 6 Gbit/s SAS/SATA Controller Product Overview*, issue 2.
3. PMC-2080174, *PM8001 Tachyon SPC 8x6G 8-Port 6-Gbit/s SAS/SATA Port Controller Hardware Specification*, issue 2.
4. PMC-2080187, *PM8001 Tachyon SPC 8x6G Ball Map*, issue 1.
5. PMC-2060785, *TISA Transport Independent Software Architecture Specification*, issue 7.
6. PMC-2061872, *sTSDK SAS/SATA Low-Level Layer Architecture Specification*, issue 14.
7. PMC-2100463, *PM8001 Tachyon SPC 8x6G Dev 01.10 Firmware Release Notes*, issue 1.

1 Introduction

The PM8001 Tachyon SPC 8x6G is a highly integrated 8-port 1.5/3/6-Gbit/s SAS/SATA controller optimized for fabric-attached storage system applications and native SAS/SATA host bus adapter (HBA) applications. The controller integrates eight 6-Gbit/s SAS-2-compliant controller ports, a high performance eight-lane PCI Express (PCIe) 2.0 compliant port, and dual multi-thread-capable MIPS 34K processors.

On the back-plane of a host bus adapter, the controller interfaces directly to an industry-standard PCIe interface. On the front-plane, each SAS port can be independently configured to use an internal 1.5, 3 or 6 Gbit SERDES.

1.1 Features

1.1.1 SAS/SATA Interface

- Integrated high performance state-machine-based eight-channel SAS/SATA 1.5G/3G/6G fully offloaded protocol controller
- Supports Serial SCSI Protocol (SSP) and Serial Management Protocol (SMP) operating as a SAS initiator or target
- Supports Serial ATA Tunneled Protocol (STP) operating as a SAS initiator
- Provides SATA host for directly connecting SATA HDD
- Hot-plug, staggered spin-up and SATA Native Command Queuing support
- Addressing to support 1024 SAS devices
- 4096 simultaneous I/O support
- Full simultaneous bandwidth up to 600 Mbyte/s per direction per PHY or 1200 Mbyte/s bi-directional per PHY
- Flexible wide-port capability supports any combination of widths up to eight serial interfaces per wide port

1.1.2 PCI Express (PCIe) Port

- Eight PCIe PHYs compliant with base specification 2.0, operating at 5 Gbit/s or 2.5 Gbit/s per link
- PCIe end-point operation
- MSI, MSI-X and INTx interrupt supported
- x1, x2, x4 and x8 lane configuration
- Support for all device (Dx) and link (Lx) PCIe power management features

1.1.3 Statistics and Performance Monitoring

- Enhanced SAS error monitoring for fault isolation.
- SAS topology monitoring for fault determination.
- In-band communication to PMC-Sierra's maxSAS expanders, multiplexers and enclosure management devices.

1.1.4 High Speed I/O

- Automatic negotiation of link speed
- Independent per-PHY selectable high-speed outputs support multiple programmable levels of pre-emphasis and output swing.
- Independent per-PHY-selectable high-speed inputs support multiple programmable levels of receive equalization.
- Integrated resistive termination and receiver AC coupling

1.1.5 Configurable Peripheral Interfaces

- Interfaces for flash memory, SRAM, and EEPROM
- 8 or 16-bit data transfers
- Four multi-master Two-wire Interfaces (TWIs) support variable bit rates up to 400 Kbit/s
- 12 General Purpose I/Os (GPIOs)

1.1.6 Physical

- Low-cost 672-pin 27 x 27 mm FCBGA package
- 1.0 mm ball pitch
- 90 nm CMOS technology

1.2 SPC 8x6G Applications

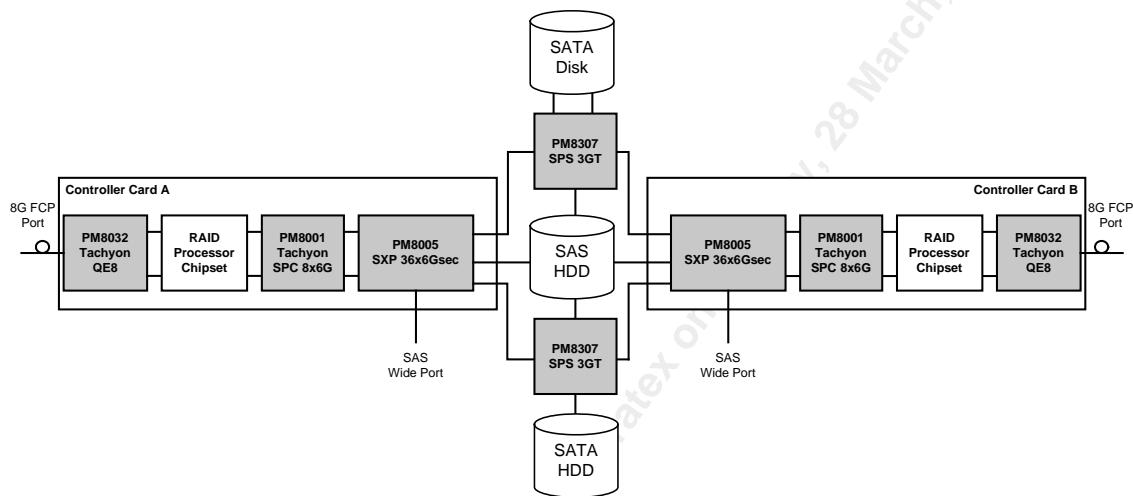
The SPC 8x6G is optimized for fabric-attached storage system applications and native SAS/SATA HBA applications capability.

1.2.1 Fabric-Attached SAS/SATA Storage System

Figure 1 shows a typical fabric-attached SAS/SATA storage system application, where the storage system is connected to a Fibre Channel network.

In each of those cases, the incoming data traffic is terminated using the appropriate protocol controller device. The terminated traffic is then passed to an x86, PowerPC or MIPS processor chipset via the PCIe interface, where storage algorithms are performed on the data. Once the processing has been completed on the data, the processor chipset passes the data to the SPC 8x6G controller, where the data is then passed to the PMC-Sierra PM8004/05 SXP 24/36x6GSec expander and PM8307 SPS 3GT multiplexer.

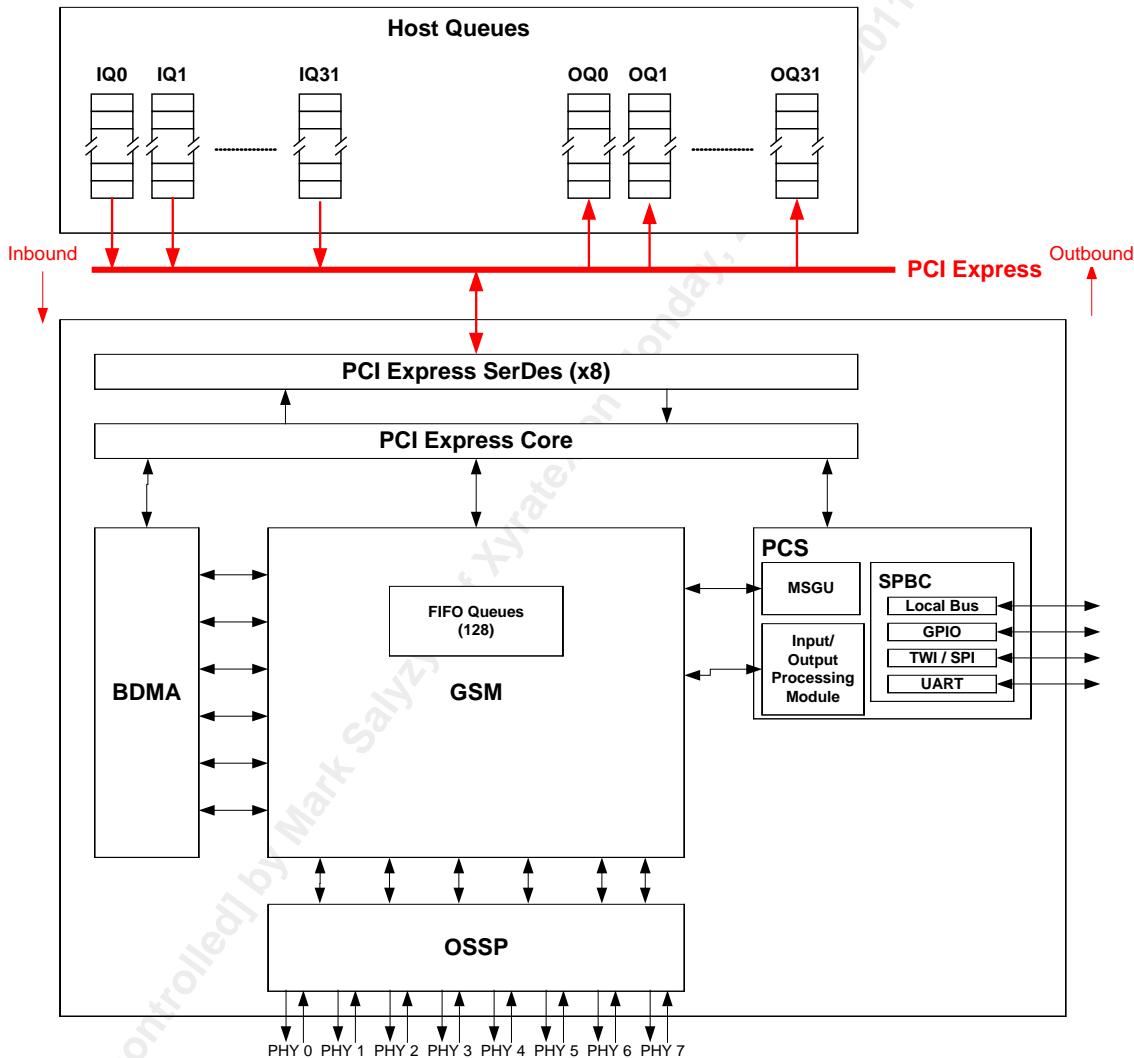
Figure 1 Fabric-Attached Storage System Application



1.3 SPC 8x6G Block Diagram

Figure 2 shows a high-level block diagram of the PM8001 SPC 8x6G controller.

Figure 2 PM8001 SPC 8x6G Block Diagram



1.4 Block Summary

This section summarizes the key device subsystems that are relevant to software.

1.4.1 Block Direct Memory Access (BDMA)

The Block Direct Memory Access (BDMA) block (located on the device PMIC block) is a multi-channel, fully programmable controller that efficiently supports 4 concurrent DMA transfers. The BDMA handles data movement among 2 address spaces: the Global Shared Memory (GSM) space and the PCIe space.

1.4.2 Global Shared Memory (GSM)

The Global Shared Memory (GSM) provides the key communication mechanism between all the blocks in the SPC 8x6G controller. The GSM is a very high-performance, ECC-protected shared memory structure with embedded logic for hardware queues and semaphores. Typical applications use the GSM to move data between blocks/interfaces, temporarily hold data, control queues, and hold program code.

The built-in FIFOs combined with the extremely high data throughput provide an efficient non-blocking mechanism to move large and small blocks of data between the device interfaces.

1.4.3 Processor Complex System (PCS)

The Processor Complex System (PCS) is composed of two RISC processors running SPC 8x6G firmware, attached fast local memory, and a Peripheral Bus Controller (PBC) subsystem implementing a variety of busses/interfaces:

- Four Two-Wire Interfaces (TWIs)
- 12 General Purpose I/Os (GPIOs) for general use and 8 for SPC 8x6G firmware

Supporting the subsystem are two firmware-implemented blocks: Message Unit (MSGU) firmware and Input/Output Processing Module firmware. The functions of these firmware blocks are summarized in the following sections.

1.4.3.1 Message Unit (MSGU)

The Message Unit (MSGU) firmware is responsible for the Message Passing Interface (MPI) with a host processor. The MSGU relays internal messages to and from different functional blocks through the PCIe interface. The MSGU also implements all host interrupt generation and interrupt coalescing policies. The MSGU firmware runs on the MIPS 34K Application Acceleration Processor (AAP1).

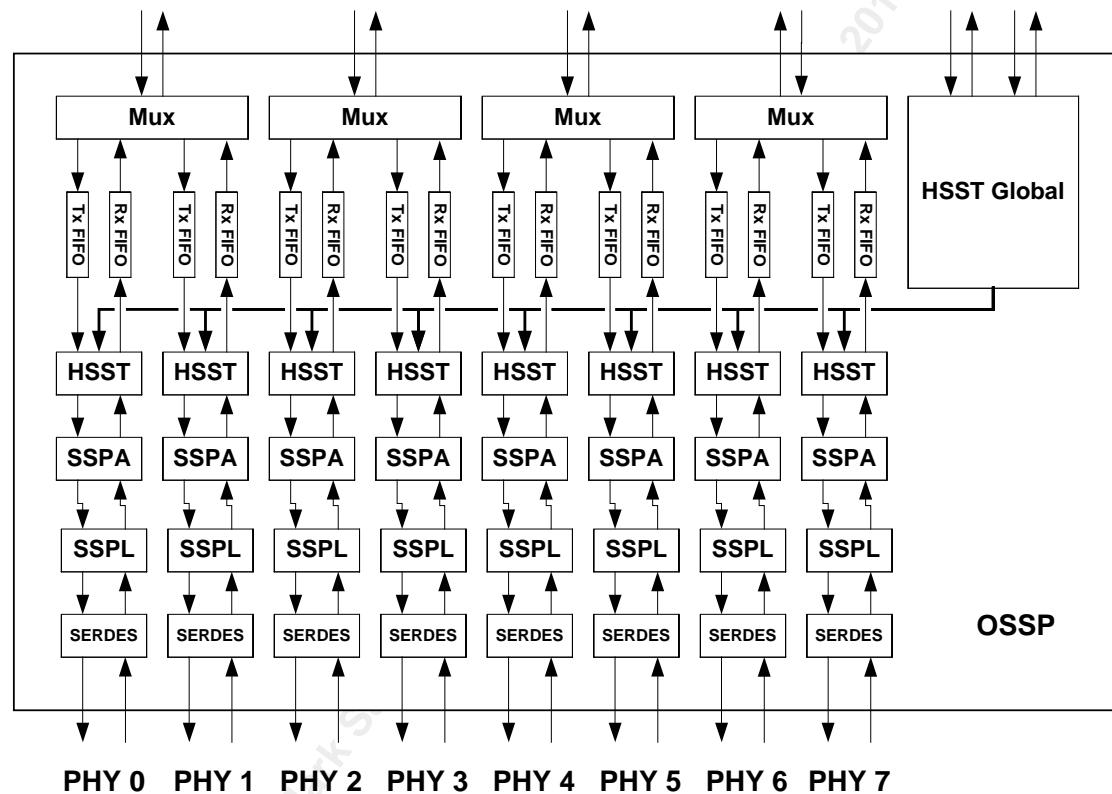
1.4.3.2 Input/Output Processing Module

The Input/Output Processing module is responsible for programming all SAS and SATA I/O operations into transport hardware state machines. The Input/Output Processing module receives I/O command messages, which are internally queued by the MSGU, and sends completion notifications and SAS/SATA event notifications back to the MSGU. The Input/Output Processing module runs on the MIPS 34K I/O processor (IOP).

1.4.4 Octal SAS/SATA Processor (OSSP) Subsystem

[Figure 3 below](#) shows the high level diagram of the Octal SAS/SATA Port (OSSP) subsystem within the SPC 8x6G controller.

Figure 3 OSSP Internal Block Diagram



The OSSP is comprised of the following:

- Hardened SAS/SATA Transport (HSST) Layer
- Global HSST Block
- SAS/SATA Port Adapter (SSPA) Link
- SAS/SATA Physical Layer (SSPL)
- SERDES Layer

1.4.4.1 Hardened SAS/SATA Transport (HSST) Layer

The Hardened SAS/SATA Transport (HSST) layer provides hardware-assisted SAS and SATA transport layer functions. It also provides full automation of SAS data transfer operations as well as SATA and SATA Tunneled Protocol (STP) data transfer using PIO, DMA and First Party DMA (FPDMA) protocols.

1.4.4.2 HSST Global Block

The HSST Global block is responsible for services required at the SAS port level involving one or more PHY links. Some functions implemented by this block are I/O context table control, internal queues control, and Transfer Ready queue control.

1.4.4.3 SAS/SATA Port Adapter (SSPA) Link

The SAS/SATA Port Adapter (SSPA) supports the SPC 8x6G 1.5G/3G/6G SAS/SATA link. The SSPA initializes the link, opens connections, transmits frames, receives frames, and closes connections. To achieve this, the SSPA implements SSP/SMP/STP/SATA Link layer functions as well as part of the port layer.

1.4.4.4 SAS/SATA Physical Layer (SSPL)

The SAS/SATA Physical Layer (SSPL) block implements the functionality of the SATA and SAS PHY layer.

1.4.4.5 Serializer/Deserializer (SERDES)

The integrated SERDES enables the OSSP to support lines rates up 1.5/3/6 Gbit/s.

1.5 Inbound Queue (IQ)

A host processor uses Inbound Queues (IQs) to submit command messages to the SPC 8x6G controller.

Messages in the IQs include data read/write requests, management requests/responses, configuration messages and control messages. See Section 7, “[Inbound Messages](#)” for more details.

1.6 Outbound Queue (OQ)

The SPC 8x6G controller uses Outbound Queues (OQs) to submit notification and response messages to a host processor.

Messages in the OQs include unsolicited inbound frames, data read/write completions, and asynchronous hardware events. See Section 8, “[Outbound Messages](#)” for more details.

2 General Programming

2.1 Endianness

2.1.1 BigEndian Format

In a Big Endian system, the “big end” is stored first; that is, for a multi-byte quantity (for example, a 32-bit integer), the Most Significant Byte (MSB) is stored first in memory. The MSB is stored in the lowest address and the Least Significant Byte (LSB) is stored in the highest address. Thus, the data is stored in memory as shown in [Table 2](#).

Table 2 BigEndian

Address 0 (MSB)				Address 1				Address 2				Address 3 (LSB)			
3 1				2 4	2 3			1 6	1 5			8 7			0

The SAS specification defines the transmission order as Big Endian; that is, the MSB is transmitted first.

2.1.2 LittleEndian Format

In a Little Endian system, the “little end” is stored first; that is, for a multi-byte quantity, the LSB is stored first in memory. The LSB is stored in the lowest address and the MSB is stored in the highest address. Thus, when data is presented on a 32-bit Little Endian bus, the bytes that form the 32-bit integer are obtained from memory as shown in [Table 3](#).

Table 3 LittleEndian

Address 3 (MSB)				Address 2				Address 1				Address 0 (LSB)			
3 1				2 4	2 3			1 6	1 5			8 7			0

The SATA specification defines the transmission order as Little Endian; that is, LSB is transmitted first.

2.1.3 Host Memory Endianness

All control structures used in the SPC 8x6G controller are stored in host memory in Little Endian format. This includes:

- Scatter/Gather Lists (SGLs)
- I/O Message Buffer (IOMB) (except SAS Information Units (IUs) and SATA FIS)

Protocol dependent payload fields in the IOMBs, like SAS Information Units or SATA FIS, are stored in the native endian format; that is, Big Endian for SAS and Little Endian for SATA.

Data buffers are treated as byte streams and are not subject to any endianness translation.

2.1.3.1 SAS Payload Endianness

The host should define the payloads of the SAS COMMAND, SAS XFER_RDY and SAS RESPONSE IUs as sequences of 32-bit integer values stored in the Big Endian format in host memory, with the MSB in the lowest address.

For example, if the Data Length field of the SSP COMMAND IU is defined as a 32-bit integer, the LSB of this field should be stored in memory at a higher address than the MSB. When this value is read from memory and transmitted by the SPC 8x6G, the lower address in memory is transmitted first, which is the MSB of the Data Length field.

Similarly, SMP COMMAND and SMP RESPONSE are in the Big Endian format.

2.1.3.2 SATA Payload Endianness

The host should define the payloads of SATA FIS as sequences of 32-bit integer values stored in the Little Endian format in host memory, with the LSB in the lowest address.

For example, if the Transfer Count field in the SATA FIS is defined as a 32-bit integer, the MSB of this field should be stored in memory at a higher address than the LSB. When this value is read from memory and transmitted by the SPC 8x6G controller, the lower address in memory is transmitted first, which is the LSB of the Transfer Count field.

2.2 Circular Queues

The SPC 8x6G controller and the host use circular queues to transfer messages and memory descriptors. These circular queues are located in host memory. Two kinds of queues are defined:

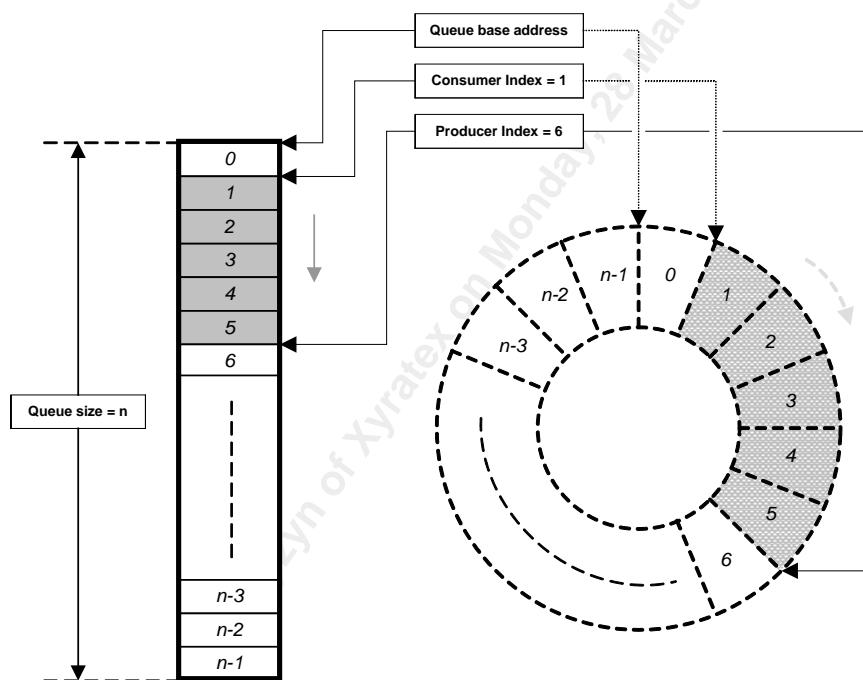
- Inbound Queues (IQs)
- Outbound Queues (OQs)

A circular queue is implemented as an array in the host memory with two associated indices, a Producer Index (PI) and a Consumer Index (CI). Both PI and CI are 32-bit entities.

As Figure 4 shows [below](#), a circular queue can be thought of as an array in a circle, rather than a straight line. In this way, as messages are added and removed from the queue, the head continually chases the tail around the array.

At different times, the queue occupies different parts of the array, but never runs out of space unless the array is fully occupied. The queue elements (queue entries) around the circle are numbered from 0 to $n-1$. When the index is moved past $n-1$, it starts over again at 0.

Figure 4 Circular Queue



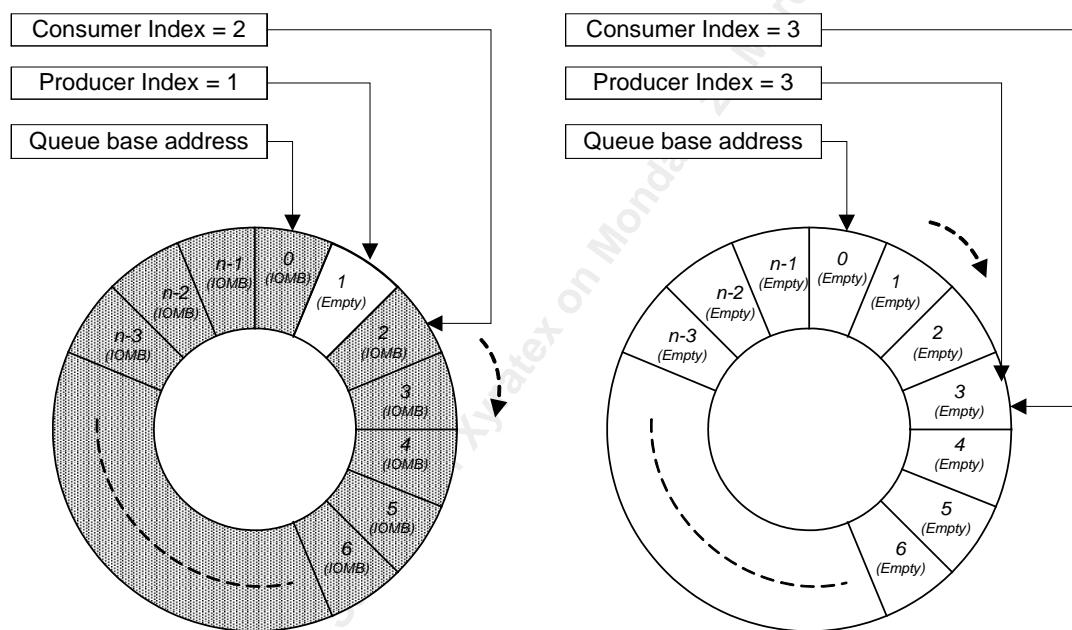
The queue is empty when the PI and CI are equal. The queue is full when the PI is exactly one position behind the CI. This method to determine the queue full/empty state always leaves an empty (reserved) position in the queue, which means that the maximum number of messages in the queue is $n-1$.

If the queue is not empty, then all queue elements from the one indicated by the CI to the element one behind the PI are “owned” (can be accessed) by the consumer. All queue elements from the one indicated by the PI to the element two behind the CI are “owned” by the producer (“two” because of the reserved empty item). When adding an element to the queue, the queue element indicated by the PI is filled and the PI is incremented (modulo n). When removing an element from the queue, the queue element indicated by the CI is emptied and the CI is incremented.

[Figure 5](#) shows the following:

- The circular queue is considered full when the producer has filled in $n-1$ entries, that is, only one entry remains empty.
- The circular queue is empty when the consumer has processed all of the entries up to the PI, and the CI equals the PI. No indication is sent to producer that the queue is empty.

Figure 5 OQ Full and Empty Example



Each circular queue can be configured to occupy a contiguous region in host memory starting a configurable “queue base address” with a size of; queue size = $n \bullet$ queue element size bytes.

2.3 Inbound Queues (IQs)

The SPC 8x6G controller uses Inbound Queues (IQs) to receive messages from the host. IQs are circular queues located in host memory. Depending on the host use model, the host can program up to 64 IQs.

IQs contain entries that describe:

- Configuration messages
- I/O Command messages

A full description of all inbound messages can be found in Section [7, “Inbound Messages”](#).

2.3.1 IQ Attributes

The number, location in host memory, size, and other attributes of the SPC 8x6G IQs are configured by the host at initialization time. The MPI Inbound Queue Configuration table (Section 5.2.3, “[MPI Inbound Queue Configuration Table Fields](#)”) shows the SPC 8x6G IQ configuration attributes. (See Section 4, “[Initialization and Configuration](#)” more details.)

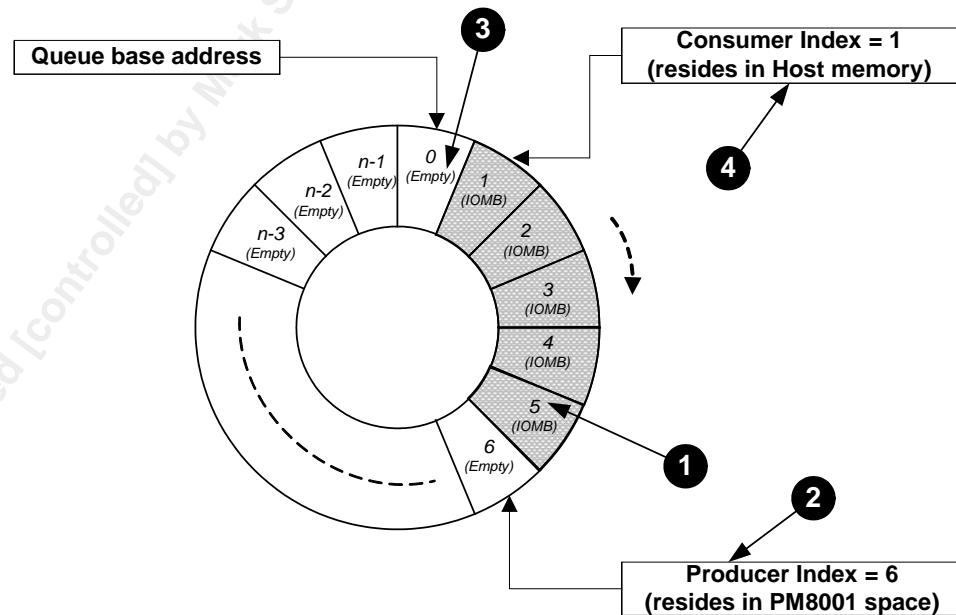
2.3.2 IQ Producer and Consumer Indexes

For IQs, the host is the producer and the SPC 8x6G controller is the consumer. The PI exists in the consumer space (the SPC 8x6G controller) and the CI exists in the producer space (host memory). This eliminates the need for any read operations of indexes access the PCIe interface, which could reduce the performance of the host I/O interface.

[Figure 6](#) shows the following:

- As the producer, the host fills the IQ entries; that is, it creates IOMBs. After the host fills in an entry, it increments and writes to the IQ PI register. The IQ PI resides in the SPC 8x6G space.
- The IQ PI points to the next empty entry the host uses to create an IOMB.
- As the consumer, the SPC 8x6G processes the IOMBs. Once the IOMB is processed, the entry is considered empty. When the SPC 8x6G processes an IOMB, it increments and writes to the IQ CI in host memory space.
- The IQ CI points to the next IOMB that the SPC 8x6G processes.

Figure 6 IQ With Five Entries Produced



Note

- The host can fill in multiple (sequential) IQ entries and then update the PI to the appropriate value in a single PCIe transaction.

2.4 Outbound Queues (OQs)

The SPC 8x6G controller uses Outbound Queues (OQs) to transfer messages to a host. OQs are circular queues located in host memory. Depending on the host use model, the host can program up to 64 OQs.

OQs contain entries that describe:

- I/O completion messages
- Hardware event notification messages
- Unsolicited frame received notification

A full description of all outbound messages can be found in Section 8, “[Outbound Messages](#)”.

2.4.1 OQ Attributes

The number, location in host memory, size and other attributes of the SPC 8x6G OQs are configured by the host at initialization time. The MPI Outbound Queue Configuration table (Section 5.2.4, “[MPI Outbound Queue Configuration Table Fields](#)” shows the SPC 8x6G OQ configuration attributes. (See Section 4, “[Initialization and Configuration](#)” for more details.)

2.4.1.1 OQ Producer and Consumer Indexes

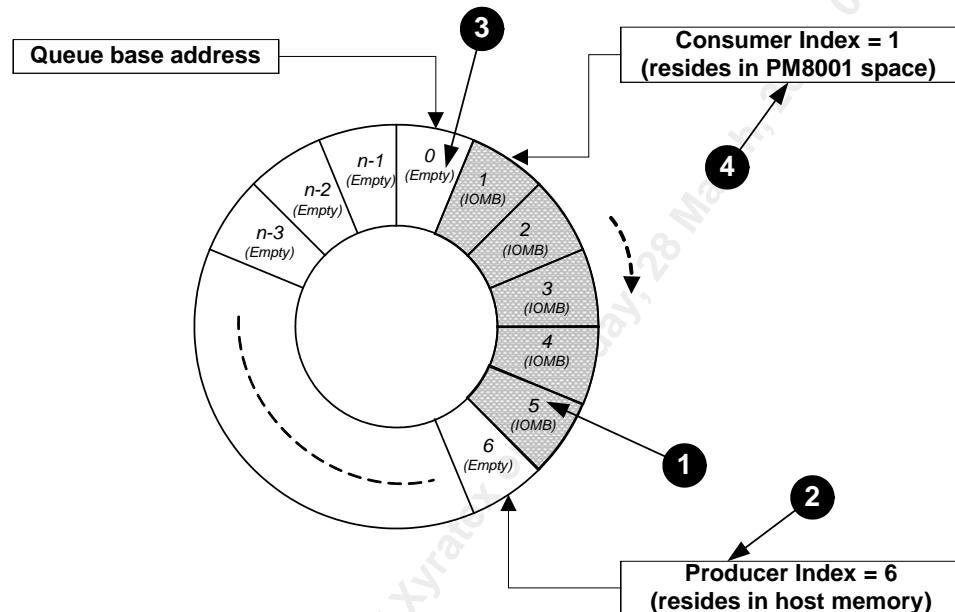
For OQs, the SPC 8x6G controller is the producer and the host is the consumer. The PI exists in consumer space (host memory) and the CI exists in the producer space (the SPC 8x6G space). This eliminates the need for any read operations of indexes across the PCIe interface, which could reduce the performance of the host I/O interface.

[Figure 7](#) shows the following:

- As the producer, the SPC 8x6G controller fills the OQ entries; that is, it creates IOMBs. After the controller fills in an entry, it increments and writes to the OQ producer index register. The OQ producer index resides in host memory. If the IOMB is larger than the OQ entry size, multiple, consecutive OQ entries are used.
- The OQ producer index points to the next empty entry the SPC 8x6G controller uses to create an IOMB.
- As the consumer, the host processes the IOMBs. Once the IOMB is processed, the entry is considered empty. When the host processes an IOMB, it increments and writes to the OQ CI in an SPC 8x6G controller register.
- The OQ CI points to the next IOMB that the host processes.

When an index points past end to the queue (the index equals the size of the OQ), it wraps to the beginning of the queue.

Figure 7 OQ With Five Entries Produced



Note

- The SPC 8x6G controller can fill in multiple (sequential) OQ entries and then update the PI to the appropriate value in a single PCIe transaction.

2.5 Scatter/Gather Lists (SGLs)

L/A pairs (Length/Address pairs) define data buffers in host memory. Each buffer (either containing outbound data to be sent or allocated to receive inbound data) is described by its length (L) in bytes, and its starting address (A). L/A pairs that define non-contiguous areas in memory are called Scatter/Gather Lists (SGLs):

- A Gather List “gathers” outbound data from host memory.
- A Scatter List “scatters” inbound data into host memory.

[Table 4](#) shows the structure of a L/A pair or SGL element.

Table 4 Scatter/Gather List Element (L/A Pair) Data Structure

	Byte 3	Byte 2	Byte 1	Byte 0
0			SGLAL	
1			SGLAH	
2			LEN	
3	E		Reserved	

Depending on the number of L/A pairs in the SGL describing the data, there are two different ways to present SGLs to the SPC 8x6G controller:

- A Local SGL
- An Extended SGL (ESGL)

Each field shown in [Table 4](#) is defined in [Table 5](#).

Table 5 SGL Element (L/A Pair) Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
0	SGL Address Low	SGLAL	4 Bytes	If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory.
1	SGL Address High	SGLAH	4 Bytes	If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory.
2	Length	LEN	4 Bytes	If Local SGL is used, this field contains the size in bytes of the data buffer in host memory. If ESGL is used (E = 1), this field is not used.
3 [31]	Extension bit	E	1 bit	This bit indicates if this SGL element (L/A pair) describes a data buffer or an ESGL in host memory. 0b: This SGL element describes a data buffer. 1b: This SGL element describes an ESGL.

2.5.1 Local SGLs

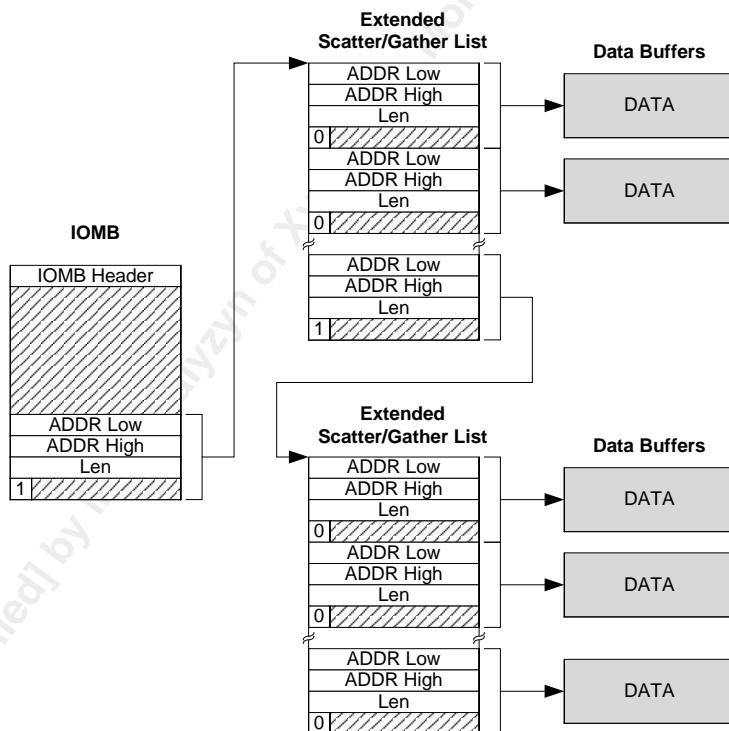
If only one pair of L/A is needed to define the data, a local SGL is used. A local SGL consists of one L/A pair and is contained entirely between the corresponding IOMB referencing the data. The advantage of using Local SGL is that fewer DMA operations are required per operation because the pair of L/A is contained within the message.

2.5.2 Extended SGLs (ESGLs)

If more than one L/A pairs are required to define the data, and ESGL is required. An ESGL is a list of L/A pairs external to the IOMB that references the data.

[Figure 8](#) shows the structure of an ESGL in host memory.

Figure 8 Extended Scatter/Gather List (ESGL)



The E field in the L/A pair contained in the IOMB is set to 1b indicating that the address fields of the L/A pair point to an ESGL in host memory with additional L/A. The SPC 8x6G does not have a specific hardware limitation on the number of L/A pairs or number of pages linked. The number of L/A pairs validated in PMC-Sierra's validation lab may have a limit arbitrarily selected based on the available test environment. This arbitrary limit may be specified in DWord 0x04 of the Configuration Table entry described in Section 5.2.1. If more L/A pairs are required, the E field of the last L/A pair of the ESGL is set to 1b and the address fields of that L/A must point to another, chained ESGL with additional L/A pairs. The total size of a SGL is arbitrary and is not limited in any way by the SPC 8x6G controller. The host may decide to use any number of ESGLs each one with a non-homogenous number of L/A pairs.

For performance considerations, the ESGL page should be 16-byte aligned and contain a multiple of four L/A pairs.

2.6 Memory Address Space

The SPC 8x6G controller provides 64 Kbytes of memory address space.

When not explicitly specified, the default base address register (BAR) settings for the PCIe interface are shown in Table 6. For configuration details see Section 8.28, “[GET_NVMD_DATA Response](#)”, Section 10, “[Chip Registers](#)”, and the soft reset recovery (normal mode and HDA mode) details in Section 11.4, “[Device Specific Fatal Error Recovery Procedures](#)”.

Table 6 Default PCIe BAR Configuration per Detected Device Type

BAR	MEMBASE	Size	Attributes	Window	AXI Address	Comment
0x10 (0)	MEMBASE-I Lower	64 KB	Memory, Non-prefetchable, ERA, 64-bit	IB_BIR_0=0	0x10_0040_0000	GSM (MSGU Reg. 1 KB)
0x14 (1)	MEMBASE-I Upper	N/A	N/A	N/A	N/A	Part of BAR 0
0x18 (2)	MEMBASE-II Lower	64 KB	Memory, Non-prefetchable, ERA, 64-bit	IB_BIR_1=2	0x10_0001_0000	GSM XCBI

BAR	MEMBASE	Size	Attributes	Window	AXI Address	Comment
0x1C (3)	MEMBASE-II Upper	N/A	N/A	N/A	N/A	Part of BAR 2
0x20 (4)	MEMBASE-III	64 KB	Memory, Non- Prefetchable 32-bit	IB_BIR_2=4	0x10_0000_0000	GSM XCBI
0x24 (5)	MEMBASE-IV	64 KB	Memory, Non- Prefetchable 32-bit	IB_BIR_3=5	0x10_004F_0000	GSM SM
0x30 (ROM)	ROMBASE	1 MB	N/A	N/A	0x18_4050_0000	Expansion ROM

2.6.1 MEMBASE-III Inbound Window Shifting

The MEMBASE-III inbound window must be shifted in order to access some of the GSM XCBI registers or GSM memory.

To shift MEMBASE-III inbound window in the GSM (AXI address 0x10_0000_0000 to 0x10_007F_FFFF), the host-side must program the [Inbound AXI Translation Lower Address – Window 2 Register](#) described in Section 10.3.12 using MEMBASE-II. The following uses AXI address 0x10_0007_E000 as an example to describe the procedure:

1. Program the [Inbound AXI Translation Lower Address – Window 2 Register](#) via MEMBASE-II to the shifted destination address. The shifted destination address is the lower 32 bits of the AXI address and must be 64-KB aligned. As an example using 0x10_0007_E000, the shifted destination address is 0x0007_0000.
2. Read back the [Inbound AXI Translation Lower Address – Window 2 Register](#) via MEMBASE-II to confirm the setting is written.
3. Using the above example, 0x10_0007_E000 can be accessed by offset 0xE000 using MEMBASE-III.

When the GSM XCBI register or GSM operation via MEMBASE-III is complete, ensure that the MEMBASE-III inbound window is mapped back to its original AXI address by completing step 1 above and setting the shifted destination address as 0x0, and step 2.

2.7 Host Buffer Alignment

For optimal performance of the SPC 8x6G, the following alignment of the inbound/outbound IOMB, host PC/CI address, event log buffers, data buffers, and extended scatter/gather list (ESGL) is recommended.

Table 7 Alignment Requirements

Data Structure	Alignment
Inbound/outbound IOMBs	32 bytes
Host PI/CI address	4 bytes
Event log buffers	32 bytes
ESGL	16 bytes
Data buffers	128 bytes

3 Functional Operation

3.1 SAS Port Instantiation, Port Context and PORT_ID

A port context represents an instance of a SAS or SATA port on the SPC 8x6G controller. The controller supports both SAS and SATA directly-attached drives and SAS or SATA devices that may be attached (sit behind) expanders directly-attached to it. Therefore, a port context may dynamically represent different kinds of ports.

When connected to other SAS end-devices or expanders, each instance of a port context represents a local SAS narrow port or wide port. When connected to a SATA device, each instance of a port context represents a local SATA port.

The host configures each SPC 8x6G PHY by specifying how it should present its link/frame sequence following the SPC 8x6G controller's initialization. See Section 4.12.1, “[PHY Initialization](#)”.

The host might assign the same SAS address to several SPC 8x6G PHYs as part of a wide port. Whether this wide port is instantiated by the controller depends on the other side's PHY configuration. If both ends have PHYs configured for a wide port (with the same SAS addresses of all PHYs on that end), then the wide port is instantiated by the controller.

The SPC 8x6G supports any combination of wide port configurations. In cases where all 8 PHYs are configured as narrow ports, the SPC 8x6G can support up to 8 different SAS addresses.

The SPC 8x6G controller identifies a SAS wide port when a SAS Identify Frame from different PHYs having the same local SAS address contains the same remote SAS address.

Port context is always instantiated by the SPC 8x6G controller. The controller instantiates a new port context following the first receipt of a SATA frame or a SAS Identify frame on each of its PHYs.

The SPC 8x6G controller provides the PORT ID to the host as part of a hardware event message. See Section 3.3, “[SAS Discovery](#)”. The host should also use PORT ID as an index to its locally-managed port context. Discovery is done on the per-port-context basis. PORT ID is a zero-based, four bit wide index. For details, see Section 8.2, “[SAS_HW_EVENT Notification](#)”.

3.2 Device Handle and DEVICE_ID

A device handle represents the relationship between an initiator and a target device. In the SCSI domain, a device handle represents the I_T nexus.

The DEVICE_ID field inside the I/O Message Buffer (IOMB) represents a device handle. The DEVICE_ID field is a 32-bit entity. The lower 16 bits assigned by the SPC 8x6G firmware are typically used for a zero-based value that the host can use as an index to a table. The upper 16 bits may be configured to contain host-specific internal information. The host should pass the entire 32-bit DEVICE_ID field to the SPC 8x6G with the lower 16 bits unmodified.

A device handle is used in both initiator and target mode operations. In both modes, the SPC 8x6G controller always instantiates the device handle and its DEVICE_ID. Only the DEVICE_ID is passed between the SPC 8x6G controller and the host.

Each device handle has a defined state to indicate whether an operation can be done using the respective device handle (DEVICE_ID). See [Table 8](#).

Table 8 Device State Definition

Device State Name	Description
DS_OPERATIONAL (0x1)	The device is in operational or normal state. All I/O requests to/from the device specified by DEVICE_ID are operational.
DS_PORT_IN_RESET (0x2)	The SPC 8x6G has set the device to this state because the port attached to the device is in PORT_IN_RESET state. See Section 11.5.2, "Local PHY Control Link/Hard Reset Handler (Host Initiated)" for the port reset state description. All I/O requests will be returned in an SSP_COMPLETION Response , SMP_COMPLETION Response , or SATA_COMPLETION Response with the STATUS IO_PORT_IN_RESET error code. See Sections 8.3 , 8.4 , and 8.9 respectively for details.
DS_IN_RECOVERY (0x3)	The host has set the device state to the recovery state. All I/O requests to the device specified by the DEVICE_ID will be returned in an SSP_COMPLETION Response , SMP_COMPLETION Response , or SATA_COMPLETION Response with the STATUS IO_DS_IN_RECOVERY error code. See Sections 8.3 , 8.4 , and 8.9 respectively for details.
DS_NON_OPERATIONAL (0x7)	The SPC 8x6G has set the device to the non-operational state. All I/O requests to the device specified by the DEVICE_ID will be returned in an SSP_COMPLETION Response , SMP_COMPLETION Response , or SATA_COMPLETION Response with the STATUS IO_DS_NON_OPERATIONAL error code. See Sections 8.3 , 8.4 , and 8.9 respectively for details.
DS_IN_ERROR (0x4)	The SPC 8x6G has set the device (currently limited to a SATA device only) to the in-error state. The first outstanding I/O will be returned in a SATA_COMPLETION Response with the STATUS IO_DS_IN_ERROR error code. Subsequent outstanding I/Os will be returned in SATA_COMPLETION Responses with the STATUS set to IO_DS_NON_OPERATIONAL. See Section 8.9 and 11.5.6.4 for details.

A device state can be changed either by the host or by the SPC 8x6G accordingly:

- The DS_NON_OPERATIONAL and DS_PORT_IN_RESET device states can only be set by the SPC 8x6G.
- The DS_IN_RECOVERY device state can only be set by the host.
- The DS_OPERATIONAL device state can be set by either the host or the SPC 8x6G.
- The DS_IN_ERROR device state can only be set by the SPC 8x6G. Currently this device state is only applicable to the SATA device as part of device recovery during head-of-line blocking of the SPC 8x6G internal Request Queue. See Section [11.5.6.4](#) for details.

As part of error recovery, the host may set the device state using the [SET_DEVICE_STATE Command](#). See Section [7.29](#), “[SET_DEVICE_STATE Command](#)” for details.

For performance reasons, the SPC 8x6G does not validate the DEVICE_ID in the I/O path. It does however, check the device state based on the lower 16 bits of the DEVICE_ID prior to acting upon the I/O command.

3.2.1 Initiator Mode DEVICE_ID

In general, in initiator mode, the instantiation of the device handle is done in association with device registration using the [REGISTER_DEVICE Command](#) (see Section [7.15](#), “[REGISTER_DEVICE Command](#)”) during discovery. To learn more about host fully managed discovery, see Section [3.3](#).

3.2.2 Target Mode DEVICE_ID

In target mode, the device handle is instantiated on the very first I/O request (during SAS OPEN), from a new initiator, received by the SPC 8x6G controller port running in target mode.

In the case that a new SAS Initiator device starts an I/O with this target device, when the first SAS_OPEN primitive is received, the HSST immediately responds back with a SAS_OPEN_ACCEPT and the SPC 8x6G instantiates a new internal device handle. Then the SPC 8x6G notifies the host of the new device handle instantiation by sending a [DEVICE_HANDLE_ARRIVED Notification](#) with the DEVICE_ID assigned to the new device (See Section [8.12](#), “[DEVICE_HANDLE_ARRIVED Notification](#)”).

The host receives and accepts (or rejects) the device handle instantiation by passing the acknowledgement to the SPC 8x6G controller by sending a [DEVICE_HANDLE_ACCEPT Command](#) (see Section [7.7](#), “[DEVICE_HANDLE_ACCEPT Command](#)”). Subsequent command/data frames will be rejected if the host decides to reject the device handle.

3.2.3 Combo Initiator and Target Mode DEVICE_ID

There is a special case where a PHY/port can be simultaneously used in both initiator and target modes:

- In initiator mode, the host registers the device by sending the [REGISTER_DEVICE Command](#) with, as an option, the host-assigned part of the DEVICE_ID. (See Section 7.15, “[REGISTER_DEVICE Command](#)” for details.)
- In target mode, the SPC 8x6G assigns the DEVICE_ID and passes it to the host using the [DEVICE_HANDLE_ARRIVED Notification](#). (See Section 8.12, “[DEVICE_HANDLE_ARRIVED Notification](#)”). When the host accepts the remote device handle, it sends a [DEVICE_HANDLE_ACCEPT Command](#) with an option to overwrite the host-assigned part of the DEVICE_ID (upper 16 bits). (See Section 7.7, “[DEVICE_HANDLE_ACCEPT Command](#)” for details.)

In the SPC 8x6G, if the PHY/port of the end node uses the same SAS address for both target and initiator modes, only one DEVICE_ID will be used.

The order in which the DEVICE_ID is initialized/instantiated in initiator mode or target mode impacts the process. *The host application is required to control the initialization sequence to provide the intended result as described in the following subsections.*

Case 1: Initiator Mode Starts First

In this case, the initiator sends a [REGISTER_DEVICE Command](#) IOMB to register the device. In the [DEVICE_REGISTRATION Response](#), the 32-bit DEVICE_ID is returned, including the host-assigned part (upper 16 bits) of the DEVICE_ID.

Later, while in target mode, the host will still receive the [DEVICE_HANDLE_ARRIVED Notification](#). If the host wishes to accept target requests from the corresponding DEVICE_ID, it must reply with the [DEVICE_HANDLE_ACCEPT Command](#) with the host assigned DEVICE_ID portion unchanged. If the host does not want to accept target requests, it must reply using the [DEVICE_HANDLE_ACCEPT Command](#) with the DEVA field set to reject. Replying with a reject in the [DEVICE_HANDLE_ARRIVED Notification](#) while in target mode does not stop initiator I/O operations with the same node (same DEVICE_ID).

In this sequence, the host, acting as a target, can reject receiving/accepting requests from the remote initiator.

Case 2: Target Mode Starts First

In this case, the host, acting in target mode, receives the [DEVICE_HANDLE_ARRIVED Notification](#). If the host wishes to accept target requests from the corresponding DEVICE_ID, it replies with a [DEVICE_HANDLE_ACCEPT Command](#) with the option to change/set the host-assigned device ID portion of DEVICE_ID.

Later, when acting as initiator, the host is not required to send the [REGISTER_DEVICE Command](#) to register the remote device, as the same DEVICE_ID has been registered in SPC 8x6G.

In this sequence, both initiator and target modes are always enabled.

3.3 SAS Discovery

SAS discovery involving the expander topology is completely managed by the host.

No discovery is needed for the directly-attached devices as they are reported during the first link up (PHY up) hardware event reported in the [SAS_HW_EVENT Notification](#). The host will still need to do device registration using the [REGISTER_DEVICE Command](#). See Section 7.15, “[REGISTER_DEVICE Command](#)”.

The host implements the SAS discovery algorithm and executes the discovery processes by sending SMP requests to the expander(s).

As the host discovers new devices, it registers the newly discovered target devices with the SPC 8x6G controller by using the [REGISTER_DEVICE Command](#). (See Section 7.15, “[REGISTER_DEVICE Command](#)”.)

One of the parameters passed during device registration is the device information/descriptor. For each target device that the host registers, the SPC 8x6G controller responds to/acknowledges the device registration by sending a [DEVICE_REGISTRATION Response](#) with the DEVICE_ID assigned to the target device being registered. (See Section 8.6, “[DEVICE_REGISTRATION Response](#)”.)

The host managed discovery process involves three main steps:

1. Obtaining the DEVICE_IDs of the direct-attached devices (SAS expanders and direct attached HDDs) by registering the directly attached device through the [REGISTER_DEVICE Command](#).
2. Sending SMP requests to the SAS expanders to determine if there are additional devices connected to them.

SMP discovery is needed only if any expanders are directly attached. The discovery process is done on a per-port basis. The SAS discovery algorithm is managed by the host and defined by the T10 SAS specification. The message command to send an SMP request is described in Section 7.13, “[SMP_REQUEST Command](#)”.

3. Registering each of the devices discovered in step (2).

Device registration is needed only if a device is found during SMP discovery. The host completes registration by sending a registration command. See Section 7.15, “[REGISTER_DEVICE Command](#)”.

The SPC 8x6G controller acknowledges the device registration with a device registration response. See Section 8.6, “[DEVICE_REGISTRATION Response](#)”. This response includes the SPC 8x6G-assigned DEVICE_ID.

The SPC 8x6G-assigned DEVICE_ID is used for all subsequent operations related to the registered device. If for some specific reason the host does not desire to use or communicate with the registered device, it may initiate deregistration of a DEVICE_ID by sending a deregistration command. See Section 7.11, “[DEREGISTER_DEVICE_HANDLE Command](#)”.

3.4 SAS Assists

The SAS standard defines the interconnect layer specified in the SCSI Architecture Model (SAM) and three transport protocols:

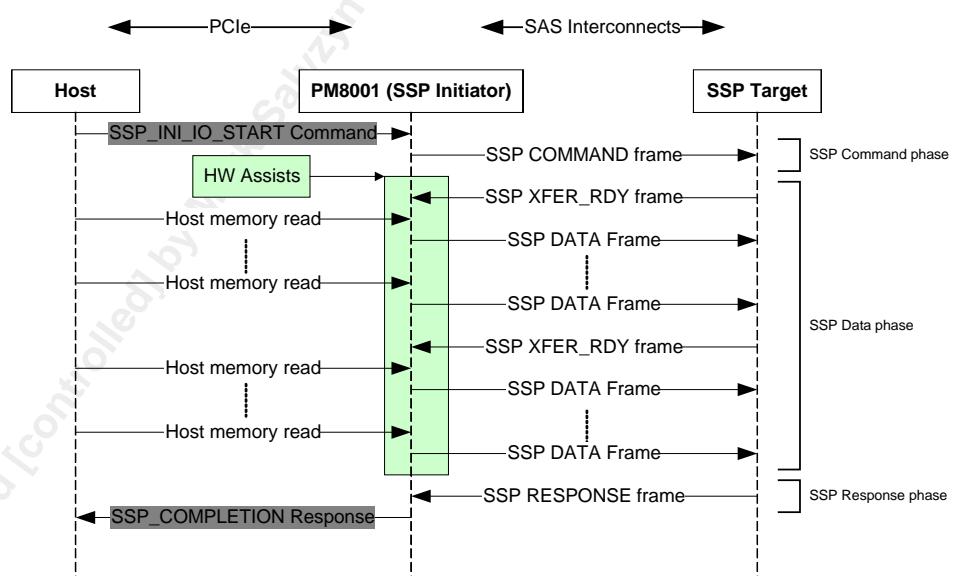
- Serial SCSI Protocol (SSP)
- Serial ATA Tunneled Protocol (STP)
- Serial Management Protocol (SMP)

The SPC 8x6G controller provides hardware assists for both SSP and STP.

3.5 SSP Initiator Write Operations

[Figure 9](#) shows the interaction flow between a host, an SPC 8x6G controller acting as an SSP initiator, and an SSP target device during a write operation.

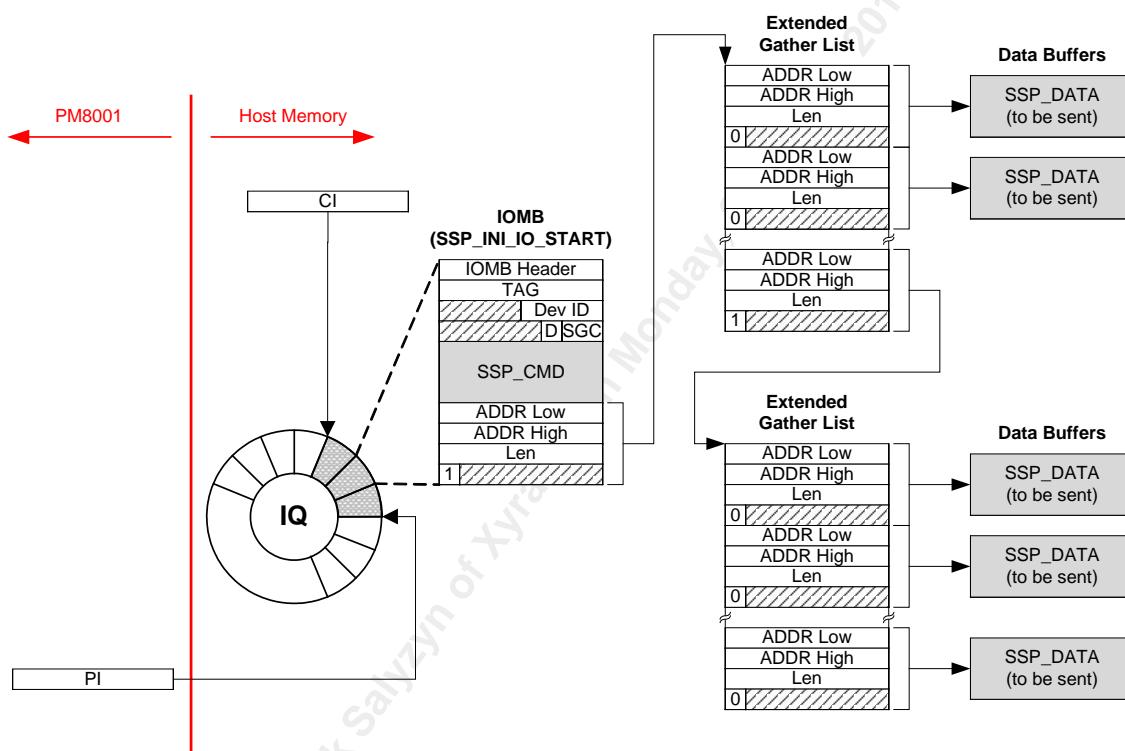
Figure 9 SSP Initiator Write Operations Flow Diagram



3.5.1 SSP Initiator Write Using an Extended Gather List

[Figure 10](#) shows the data structures required to perform a SAS write operation as an initiator where data payload requires more than one data buffer segments.

Figure 10 SSP Initiator Write Example - Extended Gather List



To perform a fully-assisted SAS initiator write operation to a target, the host selects the next available IOMB in the selected IQ.

The IOMB commands used to initiate an SSP I/O operation as an initiator are shown in [Table 9](#).

Table 9 IOMB Commands for SSP Initiator I/O Operations

IOMB Command	Page
SSP_INI_IO_START Command	211
SSP_INI_EXT_IO_START Command	218

The Extension bit (E) field of the SSP I/O start command must be set to 1b to indicate the use of an ESGL. A block of memory is then allocated as an extended gather list, the address of which is copied into the SGL Address fields: SGLAL and SGLAH.

The host then fills the addresses (SGLAL and SGLAH fields) and corresponding lengths (LEN field) of the data buffers that describe the payload for the ESGL.

The host is free to define the length of each extended gather list. If more than one extended gather list is required, it can use multiple linked gather lists. To do this, the host sets the Extension bit (E) field in the last element of each gather list and provides the address of the chained gather list in the SGLAL and SGЛАH fields. See Section 2.5, “[Scatter/Gather Lists \(SGLs\)](#)”.

The host indicates the direction of the data transfer, in this case a write operation, by writing 10b into the DIR field of the corresponding SSP I/O start command. The target device is indicated in the DEVICE_ID field. See Section 3.2, “[Device Handle and DEVICE_ID](#)”. The number of data bytes to be transferred is specified in the DL field.

The host sets the HTAG field in the corresponding SSP I/O start command to a value that references the host I/O context. This field is returned in the command completion notification message.

After the SSP_COMMAND Information Unit has been constructed in the SSP I/O start IOMB, the host instructs the SPC 8x6G controller to begin processing the I/O request. It does this by setting the Valid (V) bit in the IOMB header of the SSP I/O start command and incrementing the Producer Index (PI) of the selected IQ. After this, the SPC 8x6G controller owns the IOMB. See Section 6, “[Common IOMB Header](#)” for more details about the V bit.

The SPC 8x6G controller starts the command phase of the operation by sending an SSP COMMAND frame containing the information unit described in the SSPIU field to the target device. The target then starts the data phase of the operation by sending an SSP XFER_RDY frame. When the SPC 8x6G controller receives the SSP XFER_RDY frame, it sends the data described in the ESGL and requested by the SSP XFER_READY information unit to the target as one or more sequences of DATA frames. The target may follow each data phase with zero or more SSP XFER_RDY frames, each of which is processed in a similar manner.

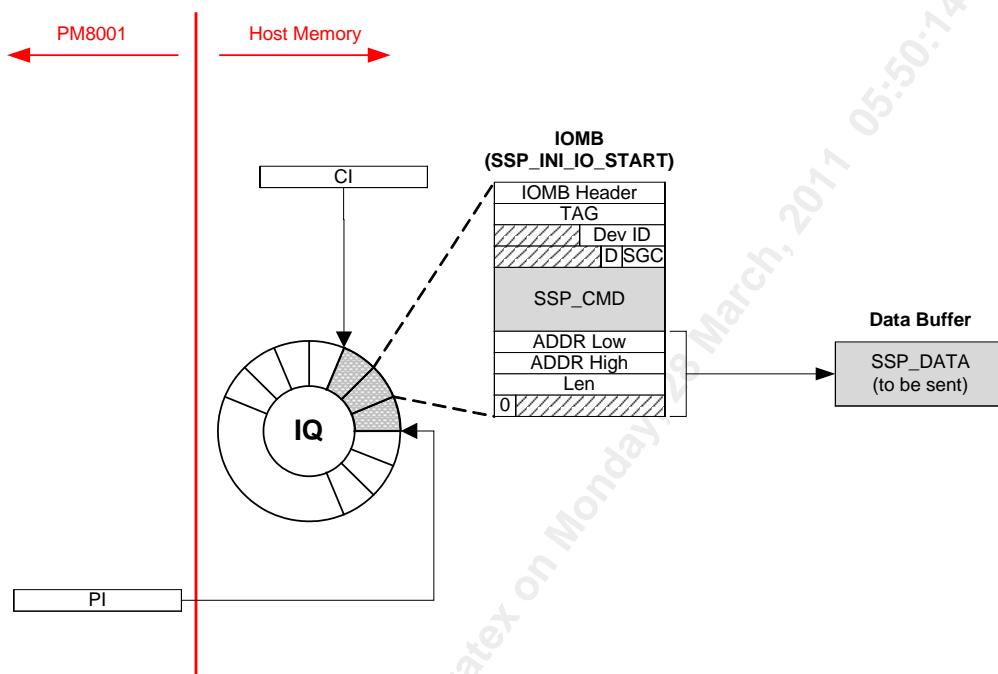
After fetching the IOMB containing the SSP I/O start command from host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)”. At this point, the host owns the IOMB so that it can be used for the next operation.

Once the target receives all of the data, it enters the response phase by sending an SSP RESPONSE frame. See Section 8.3, “[SSP_COMPLETION Response](#)”. A completion message containing the received SSP RESPONSE is sent to the OQ designated in the OBID field of the inbound IOMB command header. This signals the end of the SSP Initiator Write operation.

3.5.2 SSP Initiator Write Using a Local Gather List

[Figure 11](#) shows the data structures required to perform an SSP write operation as an initiator where data payload to be sent is in a single segment buffer.

Figure 11 SSP Initiator Write Example - Local Buffer Descriptor

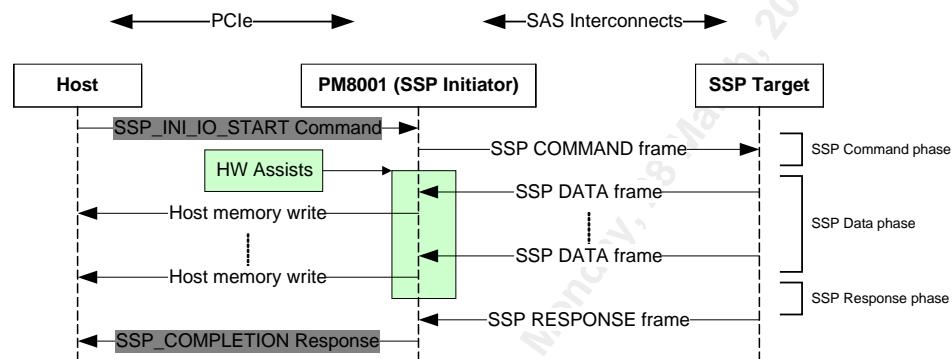


An SSP initiator write operation using a local data buffer descriptor operates in a similar manner to an extended gather list except that a single data buffer descriptor is used to describe the host memory buffer containing the data to be sent. No external list structure is required. The host fills the address (SGLAL and SGHLAH fields) and the length (LEN field) in the SSP I/O start command for the corresponding data buffer in host memory.

3.6 SSP Initiator Read Operations

[Figure 12](#) shows the interaction flow between a host, an SPC 8x6G controller acting as an SSP initiator, and an SSP target device during a read operation.

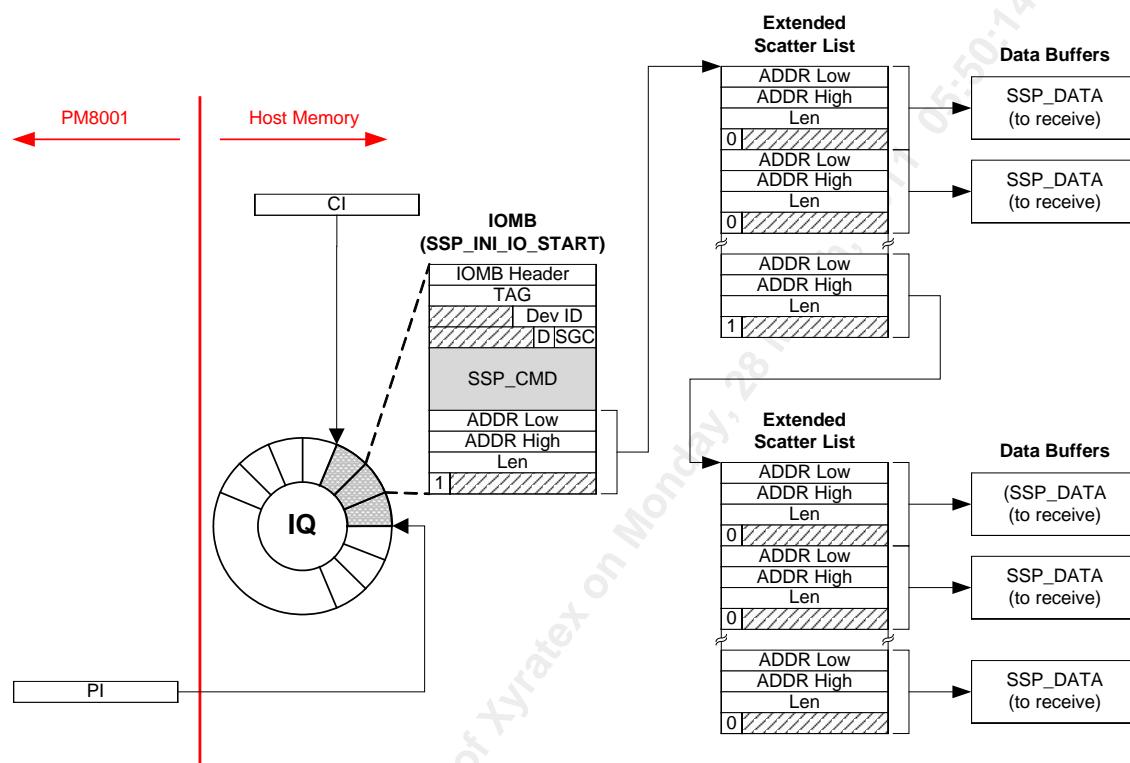
Figure 12 SSP Initiator Read Operations Flow Diagram



3.6.1 SSP Initiator Read Using an Extended Scatter List

[Figure 13](#) shows the data structures required to perform an SSP read operation as an initiator where the data to be received requires more than one data buffer segment.

Figure 13 SSP Initiator Read Example - Extended Scatter List



To perform a fully-assisted SAS initiator read operation to a target, the host selects the next available IOMB in the selected IQ. The IOMB commands used to initiate an SSP I/O operation are shown in [Table 9](#).

The Extended (E) field of the SSP I/O start command must be set to 1b to indicate the use of an ESGL. A block of memory is allocated as an extended gather list, the address of which is copied into the SGL Address fields: SGLAL and SGLAH.

The host then fills the addresses (SGLAL and SGLAH fields) and corresponding lengths (LEN field) of the data buffers that describe the payload into the ESGL.

The host is free to define the length of each extended gather list. If more than one extended gather list is required, the host can use multiple linked gather lists. To do this, the host would set the Extension bit (E) field in the last element of each gather list and provide the address of the chained gather list in the SGLAL and SGLAH fields. See Section [2.5, “Scatter/Gather Lists \(SGLs\)”](#).

The host indicates the direction of the data transfer, in this case a read operation, by writing 01b into the DIR field of the corresponding SSP I/O start command. The target device is indicated in the DEVICE_ID field. See Section [3.2, “Device Handle and DEVICE_ID”](#). The number of data bytes to be transferred is specified in the DL field.

The host sets the HTAG field in the corresponding SSP I/O start command to a value that references the host I/O context. This field is returned in the command completion notification message.

After the SSP_COMMAND Information Unit has been constructed in the SSP I/O start IOMB, the host instructs the SPC 8x6G controller to begin processing the I/O request. It does this by setting the Valid (V) bit in the IOMB header of the SSP I/O start command and incrementing the Producer Index (PI) of the selected IQ. (See Section 6, “[Common IOMB Header](#)” for details about setting the V bit.) After this, the SPC 8x6G controller owns the IOMB.

The SPC 8x6G controller starts the command phase of the operation by sending the SSP COMMAND frame containing the information unit described in the SSPIU field to the target device. The target retrieves the requested data from its media and starts the data phase of the operation by sending zero or more SSP DATA frames.

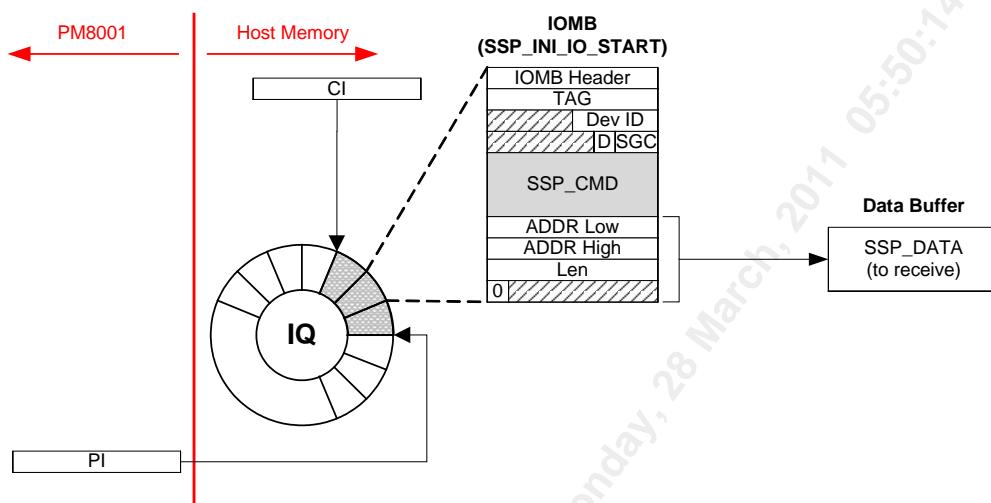
As the SPC 8x6G controller receives each SSP DATA frame, it places the data payload, as indicated by the Data Offset field of the SAS frame, into the data buffers located in host memory and defined in the ESGL.

After fetching the IOMB containing the SSP I/O start command from host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)” 73. At this point, the host owns the IOMB so that it can be used for the next operation.

Once the target receives all of the data, it enters the response phase by sending an SSP RESPONSE frame. A completion message containing the received SSP RESPONSE is sent to the OQ designated in the OBID field of the inbound IOMB command header. See Section 8.3, “[SSP_COMPLETION Response](#)”. This signals the end of the SSP Initiator Write operation.

3.6.2 SSP Initiator Read Using a Local Scatter List

[Figure 14](#) shows the data structures required to perform an SSP Read operation as an initiator where the data to be received is in a single segment buffer.

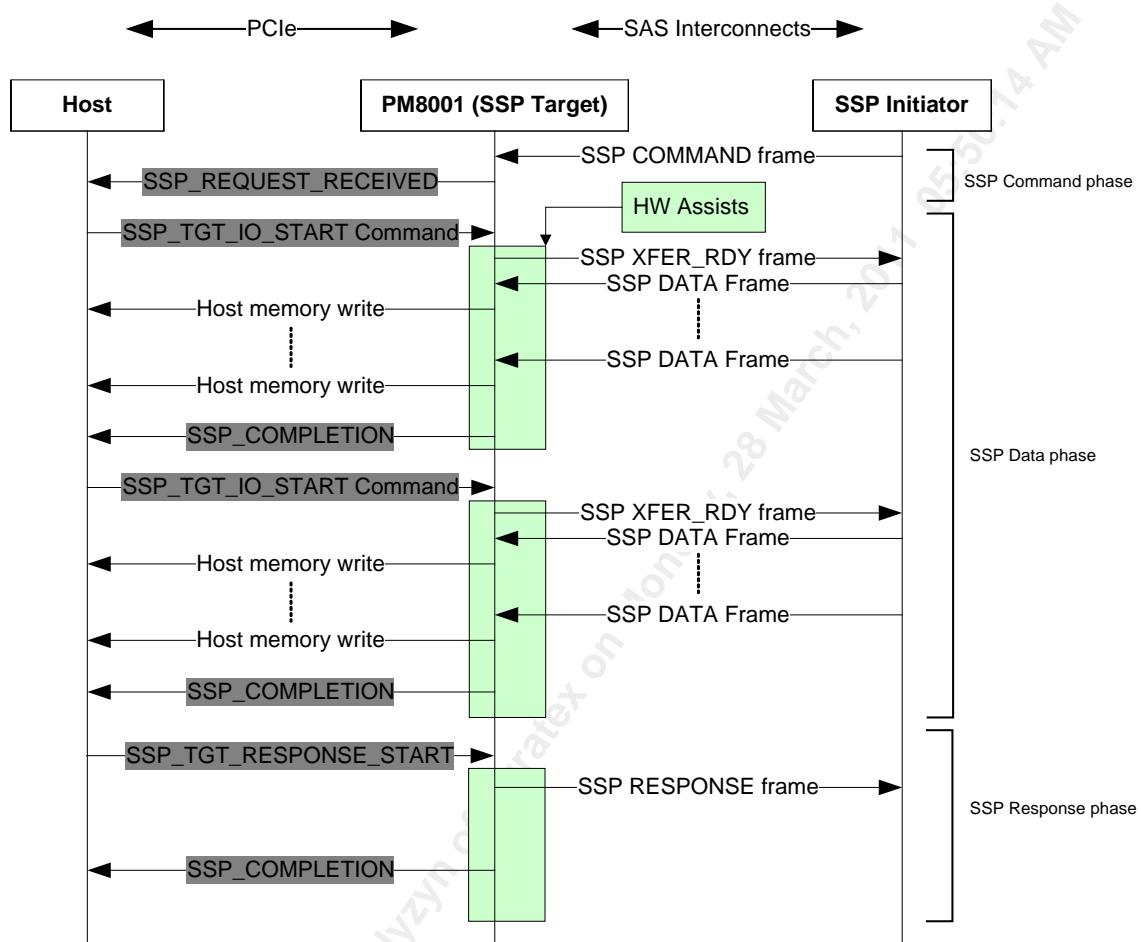
Figure 14 SSP Initiator Read Example - Local Buffer Descriptor


An SSP initiator read operation using a local data buffer descriptor is similar to an extended gather list operation. A single data buffer descriptor is used to describe the host memory buffer where the received data is stored. No external list structure is required. The host fills the address (SGLAL and SGLAH fields) and the length (LEN field) in the SSP I/O start command for the corresponding data buffer in host memory.

3.7 SSP Target Write Operations

Figure 15 shows the interaction flow between a host, an SPC 8x6G controller acting as an SSP target, and an SSP initiator device during a write operation.

Figure 15 SSP Target Write Operations Flow Diagram



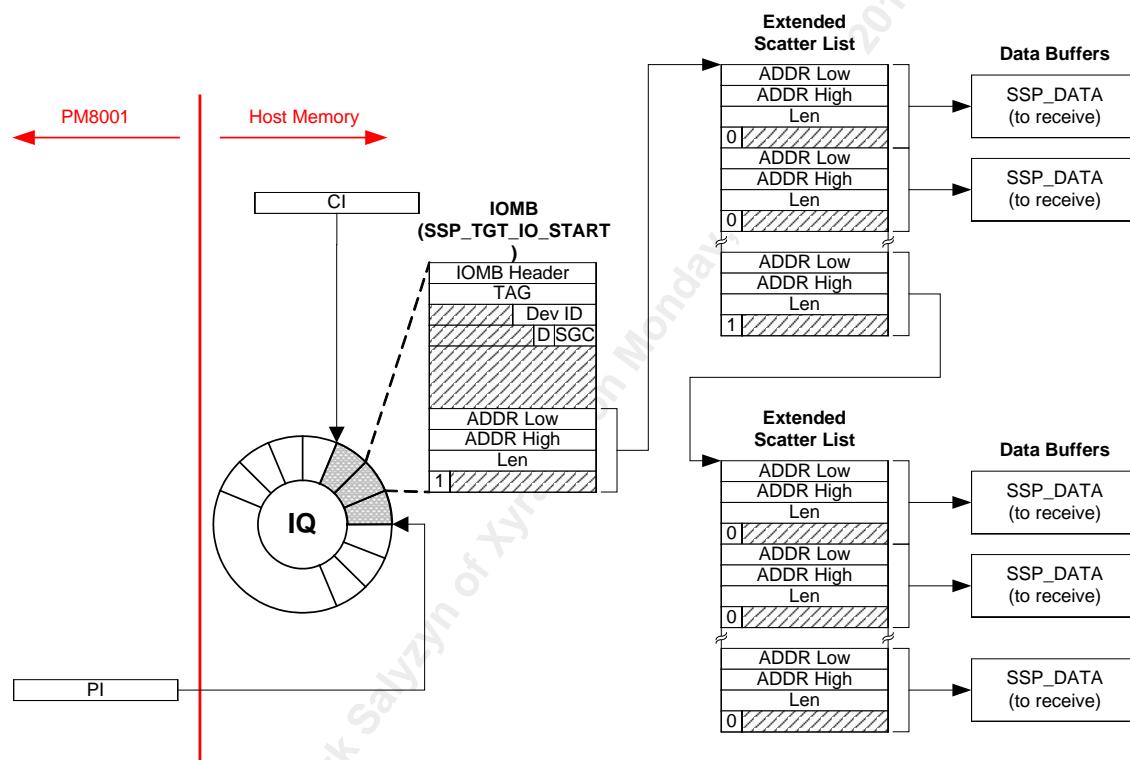
This section describes how the host initiates the data transfer phase as a target device after it receives an **SSP_COMMAND** frame requesting a write operation to this target.

Depending on the resources and the policies followed, the host may decide to complete the write data transfer operation in a single or in multiple data transfer phases.

3.7.1 SSP Target Write Using an Extended Gather List

[Figure 16](#) shows the data structures required to perform an SSP write operation as a target device where the data to be received requires more than one data buffer segment.

Figure 16 SSP Target Write Example - Extended Gather List



To perform a fully-assisted SAS target write data transfer phase operation, the host selects the next available IOMB in the selected IQ. The IOMB commands used to initiate the SSP write operation are shown in [Table 10](#).

Table 10 IOMB Commands for SSP Target I/O Operations

IOMB Command	Page
SSP_TGT_IO_START Command	222
SSP_TGT_RESPONSE_START Command	225

The Extended (E) field of the SSP I/O start command must be set to 1b to indicate the use of an ESGL. A block of memory is then allocated as an extended gather list, the address of which is copied into the SGL Address fields: SGLAL and SGLAH.

The host then fills the addresses (SGLAL and SGLAH fields) and the corresponding lengths (LEN field) of the data buffers that describe the payload for the ESGL.

The host is free to define the length of each extended gather list. If more than one extended gather list is required, the host can use multiple linked gather lists. To do this, the host sets the Extension bit (E) field in the last element of each gather list and provides the address of the chained gather list in the SGLAL and SGLAH fields. See Section 2.5, “[Scatter/Gather Lists \(SGLs\)](#)”.

The host indicates the direction of the data transfer, in this case a write operation, by writing 01b into the DIR field of [SSP_TGT_IO_START Command](#). The initiator device is indicated in the DEVICE_ID field. See Section 3.2, “[Device Handle and DEVICE_ID](#)”. The number of data bytes to be transferred is specified in the DL field. The initial offset for this data transfer phase into the overall I/O data transfer is indicated in the DO field. For a single data transfer phase I/O and the first data phase of a multiple data phase I/O the DO field is set to 0. (See Section 7.8, “[SSP_TGT_IO_START Command](#)”).

The host sets the HTAG field in [SSP_TGT_IO_START Command](#) to a value that references the host I/O context. This field is returned in the [SSP_COMPLETION Response](#) message.

After the SSP I/O start IOMB has been constructed, the host instructs the SPC 8x6G controller to begin processing the I/O request. It does this by setting the Valid (V) bit in the IOMB header of the SSP I/O start command and incrementing the Producer Index (PI) of the selected IQ. After this, the SPC 8x6G controller owns the IOMB. For more details about the V bit, see Section 6, “[Common IOMB Header](#)”.

Because this is a target command and the data transfer direction has been specified from initiator to target, the SPC 8x6G controller starts the data transfer phase by sending the SSP XFER_READY frame to the initiator device. The initiator starts the data transfer phase of the operation by sending one or more SSP DATA frames.

As the SPC 8x6G controller receives each SSP DATA frame, it places the data payload, as indicated by the Data Offset field of the SAS frame, into the data buffers located in host memory and defined in the ESGL.

After fetching the IOMB containing the SSP I/O start command form host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)”. At this point, the host owns the IOMB so it can be used for the next operation.

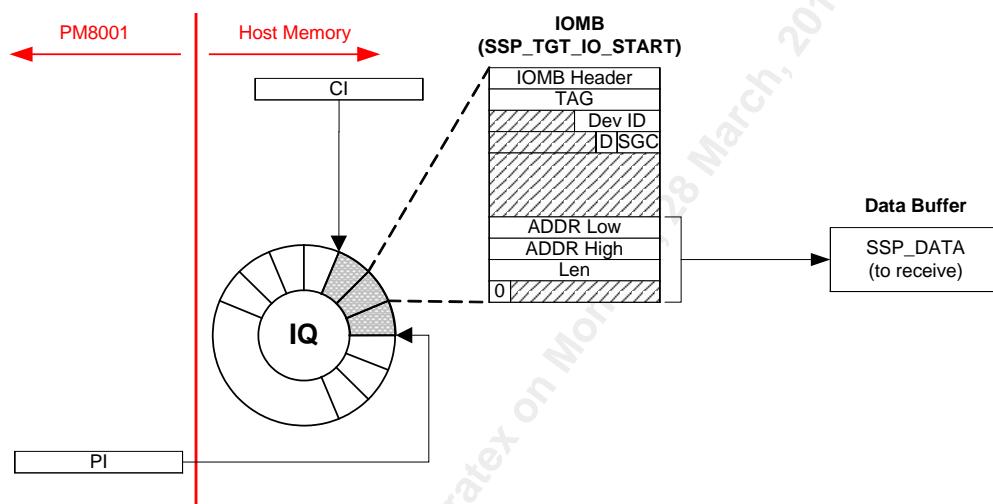
Once the data transfer phase is completed, the SPC 8x6G controller sends an [SSP_COMPLETION Response](#) message to the OQ designated in the OBID field of the inbound IOMB command header. (See Section 8.3, “[SSP_COMPLETION Response](#)”.) This signals the end of the SSP Target Write data transfer operation.

After the completion of the last data phase of the I/O, the host sends the SSP response using the [SSP_TGT_RESPONSE_START Command](#). (See Section 7.9, “[SSP_TGT_RESPONSE_START Command](#)”.)

3.7.2 SSP Target Write Using a Local Gather List

[Figure 17](#) shows the data structures required to perform an SSP write operation as a target device where the data to be received is in a single segment buffer.

Figure 17 SSP Target Write Example - Local Gather List

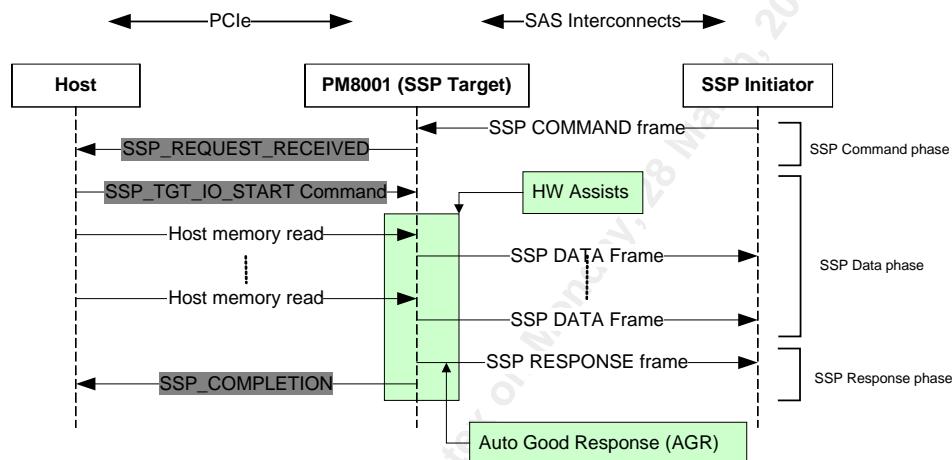


An SSP target device write operation using a local data buffer descriptor operates in a similar manner as an extended gather list. In this case, a single data buffer descriptor is used to describe the host memory buffer where the received data is stored. No external list structure is required. The host fills the address (SGLAL and SGLAH fields) and the length (LEN field) in the SSP I/O start command for the corresponding data buffer in host memory.

3.8 SSP Target Read Operations

Figure 18 shows the interaction flow between a host, an SPC 8x6G controller acting as an SSP target, and an SSP initiator device during a read operation.

Figure 18 SSP Target Read Operation Flow Diagram

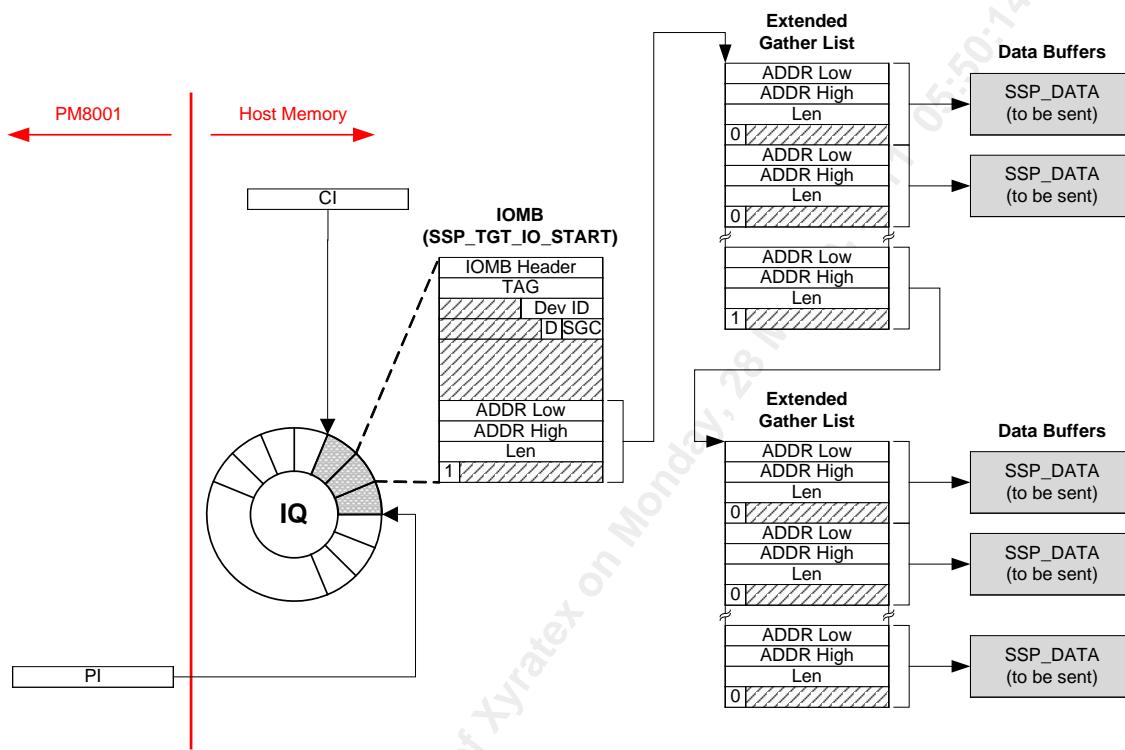


This section describes how the host initiates a data transfer phase as a target device after it receives an SSP_COMMAND frame requesting a read operation from this target.

3.8.1 SSP Target Read Using an Extended Scatter List

Figure 19 shows the data structures required to perform an SSP read operation as a target device where data payload to be sent requires more than one data buffer segment.

Figure 19 SSP Target Read Example - Extended Scatter List



To perform a fully-assisted SAS target read data transfer phase operation the host selects the next available IOMB in the selected IQ. The IOMB commands used to initiate an SSP write operation are shown in [Table 10](#).

The Extended (E) field of the SSP I/O start command must be set to 1b to indicate the use of an ESGL. A block of memory is then allocated as an extended gather list, the address of which is copied into the SGL Address fields: SGLAL and SGLAH.

The host then fills the addresses (SGLAL and SGLAH fields) and corresponding lengths (LEN field) of the data buffers into the ESGL.

The host is free to define the length of each extended gather list. If more than one extended gather list is required, it can use multiple linked gather lists. To do this, the host sets the Extension bit (E) field in the last element of each gather list and provides the address of the chained gather list in the SGLAL and SGLAH fields. See Section [2.5, “Scatter/Gather Lists \(SGLs\)”](#).

The host indicates the direction of the data transfer, in this case a read operation, by writing 10b into the DIR field of the corresponding SSP I/O start command. The initiator device is indicated in the DEVICE_ID field. See Section 3.2, “[Device Handle and DEVICE_ID](#)”. The number of data bytes to be transferred is specified in the DL field. The initial offset for this data transfer phase to the overall I/O data transfer is indicated in the DO field. For a single data transfer phase I/O and the first data phase of a multiple data phase I/O the DO field is set to 0.

The host sets the HTAG field in the corresponding SSP I/O start command to a value that references the host I/O context. This field is returned in the [SSP_COMPLETION Response](#) message described in Section 8.3.

After the SSP I/O start IOMB has been constructed, the host instructs the SPC 8x6G controller to begin processing the I/O request. It does this by setting the Valid (V) bit in the IOMB header of the SSP I/O start command and incrementing the Producer Index (PI) of the selected IQ. After this the SPC 8x6G controller owns the IOMB. See Section 6, “[Common IOMB Header](#)” for more details about the IOMB header.

Because this is a target command and the data transfer direction has been specified from target to initiator, the SPC 8x6G controller starts the data transfer phase by sending the requested data with a series of one or more SSP DATA frames to the initiator device.

As the SPC 8x6G controller sends each SSP DATA frame, it retrieves the data payload, as indicated by the Data Offset field of the SAS frame, from the data buffers located in host memory and defined in the ESGL.

After fetching the IOMB containing the SSP I/O start command from host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)”. At this point, the host owns the IOMB so it can be used for the next operation.

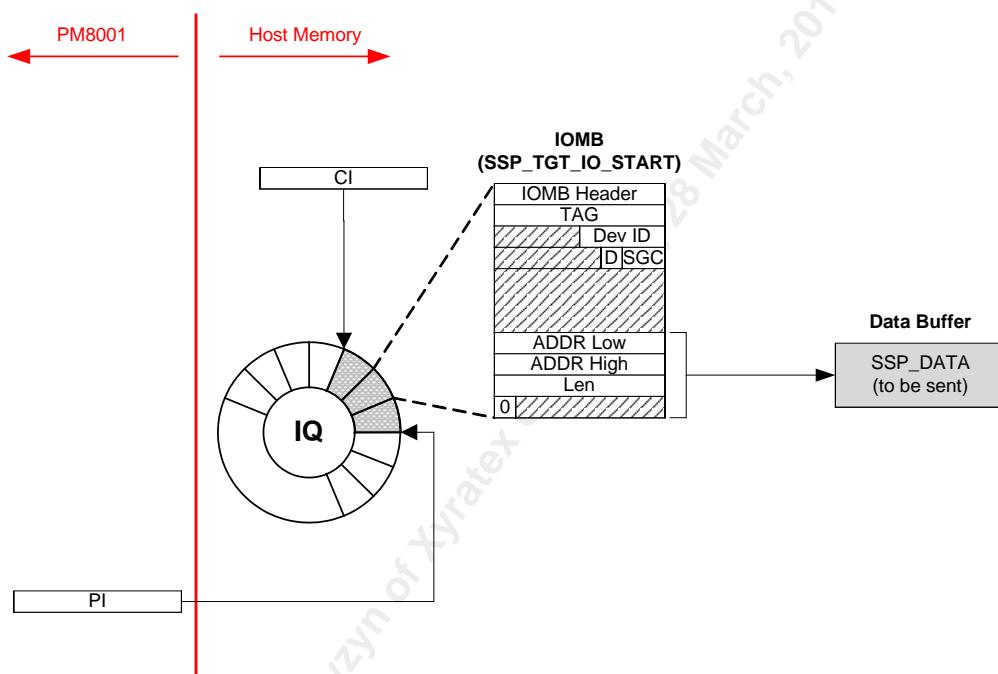
Once the data transfer phase is completed, the SPC 8x6G controller sends an [SSP_COMPLETION Response](#) message to the OQ designated in the OBID field of the inbound IOMB command header. (See Section 8.3, “[SSP_COMPLETION Response](#)”.) This signals the end of the SSP Target Write data transfer operation.

If this is the single or the last data transfer phase for the I/O operation the host can set the Auto Good Response (AGR) field in the SSP I/O start command. This instructs the SPC 8x6G controller to automatically send an SSP RESPONSE frame to the initiator device with a good status code. See Section 7.8, “[SSP_TGT_IO_START Command](#)”.

3.8.2 SSP Target Read Using a Local Scatter List

[Figure 20](#) shows the data structures required to perform an SSP read operation as a target device where data payload to be sent is in a single segment buffer.

Figure 20 SSP Target Read Example - Local Scatter List



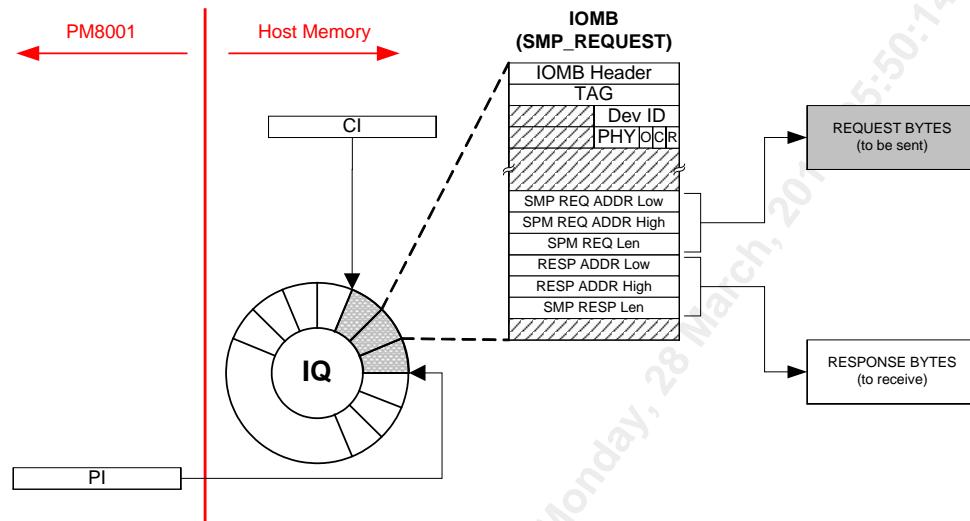
An SSP target device read operation using a local data buffer descriptor operates in a similar way as with an extended gather list except. In this case, a single data buffer descriptor is used to describe the host memory buffer containing the data to be sent. No external list structure is required. The host fills the address (SGLAL and SGЛАH fields) and the length (LEN field) in the SSP I/O start command for the corresponding data buffer in host memory.

3.9 SMP Request Operations

3.9.1 SMP Request Using Extended Request and Response Buffers

[Figure 21](#) shows the data structures required to perform an SMP initiator operation where the **SMP_REQUEST** bytes are specified in an external buffer and the **SMP_RESPONSE** is stored in the specified memory buffer.

Figure 21 SMP Request Example - Extended Request and Response Byte Buffers



To perform an SMP initiator operation to a target with extended request and response buffers, the host selects the next available IOMB in the selected IQ to construct the [SMP_REQUEST Command](#) used to initiate an SMP operation. (See Section 7.13, “[SMP_REQUEST Command](#)”.)

The host then allocates a block in the host memory to construct the SMP REQUEST frame to be sent. The host writes its address into the ISPAL and ISPAH fields and its size in the ISPL field of the [SMP_REQUEST Command](#).

The host then sets the Indirect Payload (IP) field in the [SMP_REQUEST Command](#) to indicate that the ISPAL, ISPAH, and ISPL are valid fields.

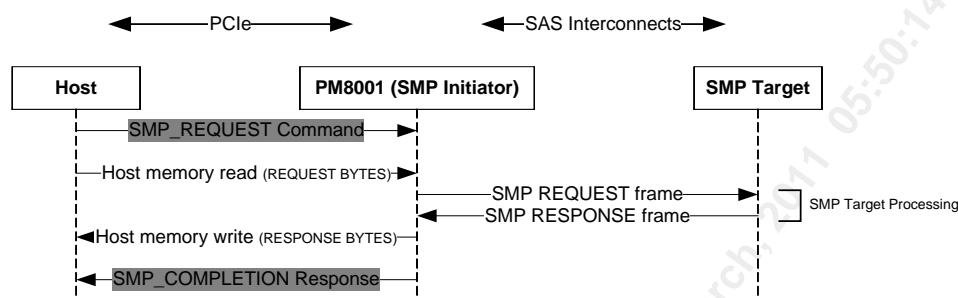
The host also allocates a host memory buffer to put the received SMP RESPONSE frame and writes its address into the ISRAL and ISRAH fields and its size into ISRL field in the [SMP_REQUEST Command](#).

The host then sets the Indirect Response (IR) field in the [SMP_REQUEST Command](#) to indicate that the ISRAL, ISRAH, and ISRL are valid fields.

The host can specify the PHY to be used to perform the SMP operation by setting the Override (OV) field and specifying the PHY Identifier in the PHYID field.

The host sets the HTAG field in the [SMP_REQUEST Command](#) to a value that references the host command context. This field is returned in the [SMP_COMPLETION Response](#) message. (See Section 8.4, “[SMP_COMPLETION Response](#)” for details.)

Figure 22 SMP Initiator with Extended Request and Response Buffers Flow Diagram



The host instructs the SPC 8x6G controller to begin processing the SMP request. It does this by setting the Valid (V) bit in the IOMB header and incrementing the Producer Index (PI) of the selected IQ. After this the SPC 8x6G controller owns the IOMB. See Section 6, “[Common IOMB Header](#)” for more details about the V bit.

As shown in [Figure 22](#), the SPC 8x6G controller sends the SMP REQUEST frame with the specified SMP REQUEST frame bytes to the target device. If PHY override was not specified in the [SMP_REQUEST Command](#), the SPC 8x6G controller selects the next available PHY in the port based on its internal scheduling algorithm.

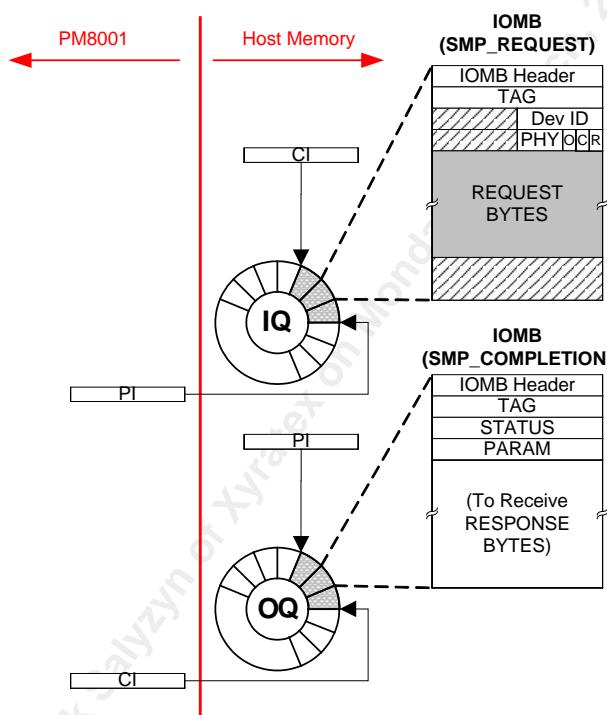
After fetching the IOMB containing the [SMP_REQUEST Command](#) from host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)”. At this point, the host owns the IOMB so it can be used for the next operation.

The corresponding PHY remains open until the SMP RESPONSE frame sent by the target device is received. The SPC 8x6G controller writes the received RESPONSE bytes into the receive buffer in host memory specified in the [SMP_REQUEST Command](#). The SPC 8x6G controller then posts a [SMP_COMPLETION Response](#) message into the OQ specified in the OBID field of the IOMB command header. (See Section 8.4, “[SMP_COMPLETION Response](#)” for details.)

3.9.2 SMP Request Using Local (Embedded) Request and Response

Figure 23 shows the data structures required to perform an SMP initiator operation where the SMP REQUEST frame bytes are embedded in the [SMP_REQUEST Command IOMB](#) and the SMP RESPONSE frame bytes are embedded in the [SMP_COMPLETION Response IOMB](#).

Figure 23 SMP Request Example – with Local Request and Response Bytes



To perform an SMP initiator operation to a target with local request and response buffers, the host selects the next available IOMB in the selected IQ to construct the [SMP_REQUEST Command](#) used to initiate an SMP operation. (See Section 7.13, “[SMP_REQUEST Command](#)”.)

The host includes the SMP REQUEST frame bytes as part of the [SMP_REQUEST Command](#).

The [SMP_REQUEST Command](#) can hold up to 48 SMP REQUEST bytes.

Note

- The local SMP request mechanism allows SMP REQUEST frames to be sent with up to 48 REQUEST bytes excluding the CRC bytes, which are generated automatically by hardware.

The host then clears the Indirect Payload (IP) field in the [SMP_REQUEST Command](#) to indicate that the DWords 8 to 11 of the [SMP_REQUEST Command](#) contain the SMP REQUEST frame bytes 16 to 31.

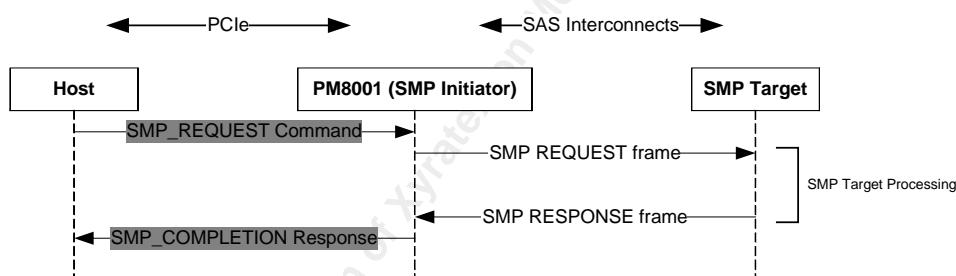
The host also clears the Indirect Response (IR) field to indicate that the SMP RESPONSE frame bytes are to be included as part of the [SMP_COMPLETION Response](#) IOMB. (See Section 8.4, “[SMP_COMPLETION Response](#)”.)

The host can specify the PHY that is used to perform the SMP operation by setting the Override (OV) field and specifying the PHY Identifier in the PHYID field.

The host sets the HTAG field in the [SMP_REQUEST Command](#) to a value that references the host command context. This field is returned in the [SMP_COMPLETION Response](#) message.

The host instructs the SPC 8x6G controller to begin processing the SMP request. It does this by setting the Valid (V) bit in the IOMB header of the [SMP_REQUEST Command](#) and incrementing the Producer Index (PI) of the selected IQ. After this, the SPC 8x6G controller owns the IOMB. See Section 6, “[Common IOMB Header](#)” for more details about the V bit.

Figure 24 SMP Initiator with Local Request and Response Bytes Flow Diagram



As shown in [Figure 24](#), the SPC 8x6G controller sends the SMP REQUEST frame with the specified SMP REQUEST frame bytes to the target device. If a PHY override was not specified in the [SMP_REQUEST Command](#), the SPC 8x6G controller selects the next available PHY in the port based on its internal scheduling algorithm.

After fetching the IOMB containing the [SMP_REQUEST Command](#) from host memory, the SPC 8x6G controller clears the Valid (V) bit in the IOMB header and increases the Consumer index (CI) of the corresponding IQ. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)”. At this point, the host owns the IOMB so it can be used for the next operation.

The corresponding PHY remains open until the SMP RESPONSE frame sent by the target device is received. The SPC 8x6G controller posts a [SMP_COMPLETION Response](#) IOMB into the OQ specified in the OBID field of the inbound IOMB command header. The controller then includes the received SMP RESPONSE bytes as part of the [SMP_COMPLETION Response](#). (See Section 8.4, “[SMP_COMPLETION Response](#)”.)

3.10 SSP Initiator Task Management Operation

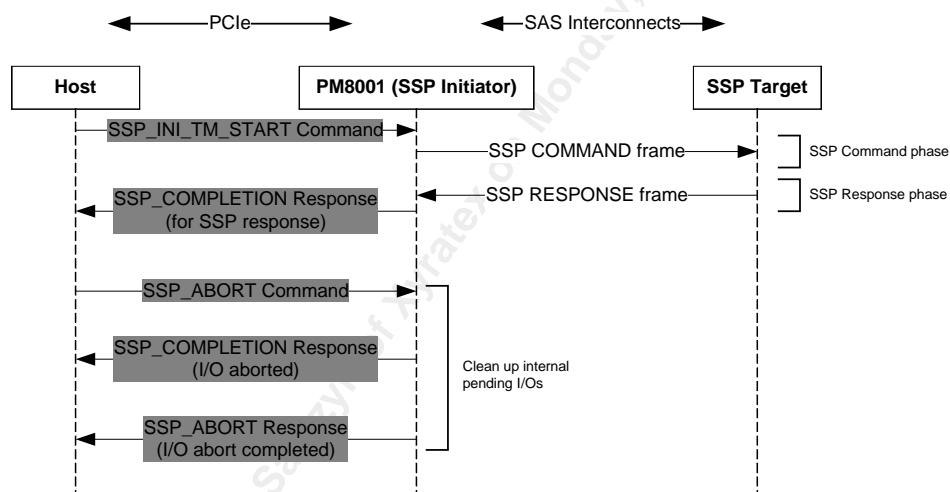
[Figure 25](#) shows the interaction flow between a host, an SPC 8x6G controller acting as an SSP initiator, and an SSP target device during a task management operation.

The host cleans up any outstanding I/Os inside the SPC 8x6G by issuing an [SSP_ABORT Command](#). (See Section 7.10, “[SSP_ABORT Command](#)” for details.)

For a task management operation such as ABORT_TASK, ABORT_TASK_SET, CLEAR_TASK_SET, and LOGICAL_UNIT_RESET, the host must send an [SSP_ABORT Command](#) for each outstanding I/O impacted by the task. (That is, for a Logical Unit Reset, an [SSP_ABORT Command](#) must be sent for all pending I/Os associated with that Logical Unit Number (LUN)).

For the Query management task, the host must send an [SSP_ABORT Command](#) if query returns a FUNCTION COMPLETE (that is, the specified task is not present in the target task set).

Figure 25 SSP Initiator Task Management Flow Diagram



The [SSP_INI_TM_START Command](#) contains options to allow some control for handling outstanding I/Os to the device when a task management command is sent to the target device. See Section 7.5, “[SSP_INI_TM_START Command](#)”, Section 7.29, “[SET_DEVICE_STATE Command](#)” and Section 11.5.6.2, “[Task Management Special Control with Device State](#)”.

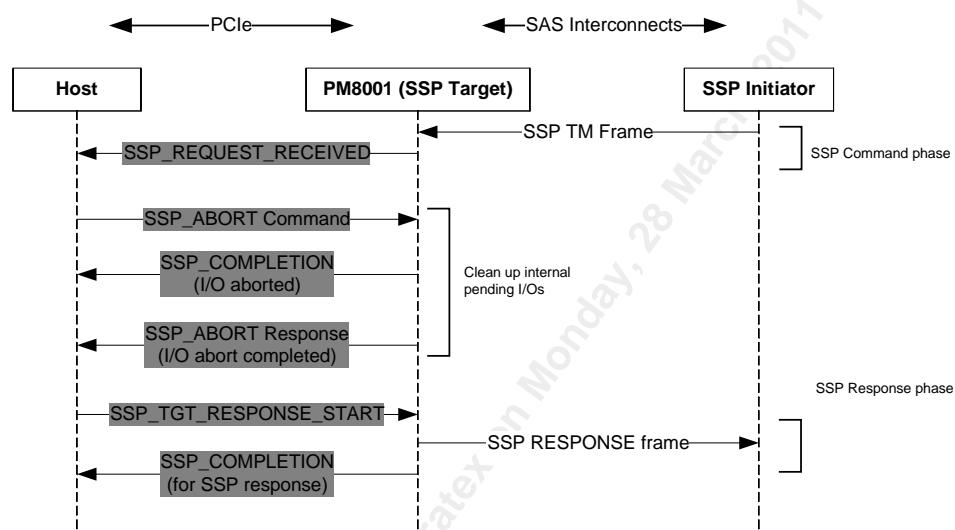
3.11 SSP Target Task Management Operation

[Figure 26](#) shows the interaction flow between a host, an SPC 8x6G controller acting as an SSP target, and an SSP initiator device during a task management operation.

The host is responsible in the cleaning up any outstanding I/Os inside SPC 8x6G by issuing an [SSP_ABORT Command](#). (See Section 7.10, “[SSP_ABORT Command](#)”.) This is typically the case for a Logical Unit Reset task management operation as the host needs to send an [SSP_ABORT Command](#) for each outstanding I/O associated with that LUN.

The [SSP_TGT_RESPONSE_START Command](#) should only be sent after all internal clean up is completed. (See Section 7.9, “[SSP_TGT_RESPONSE_START Command](#)” for details about this command.)

Figure 26 SSP Target Task Management Operation Flow Diagram



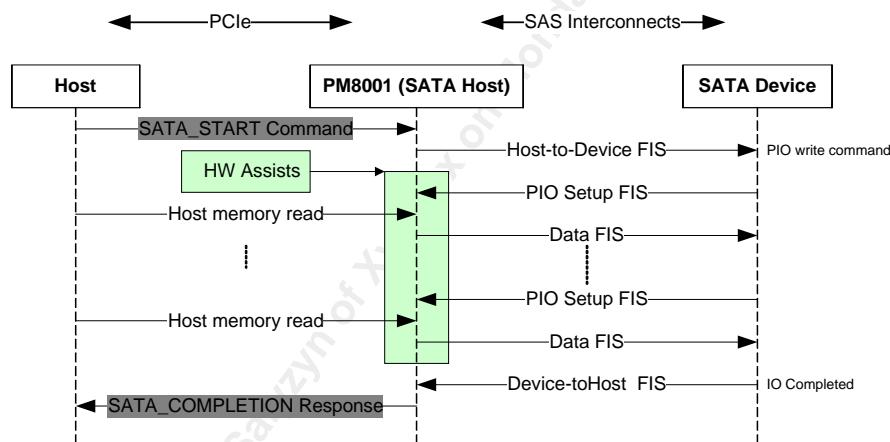
3.12 SATA Assists

The SATA protocol provides a hardware assist for data transfer operations to SATA devices. SATA devices can be directly attached to the SPC 8x6G controller's ports or to attached SAS expanders through a SATA Tunneling Protocol (STP) bridge. The SPC 8x6G controller supports PIO, DMA, and First Party DMA (FPDMA).

3.13 SATA Host Write Operations

[Figure 27](#) shows the interaction flow between a host, an SPC 8x6G controller acting as a SATA host, and a SATA device during a write operation using PIO data transfer mode.

Figure 27 Host Write Operation Flow Diagram (PIO Data Transfer Mode)



[Figure 28](#) shows the interaction flow between a host, an SPC 8x6G controller acting as a SATA host, and a SATA device during a write operation using DMA data transfer mode.

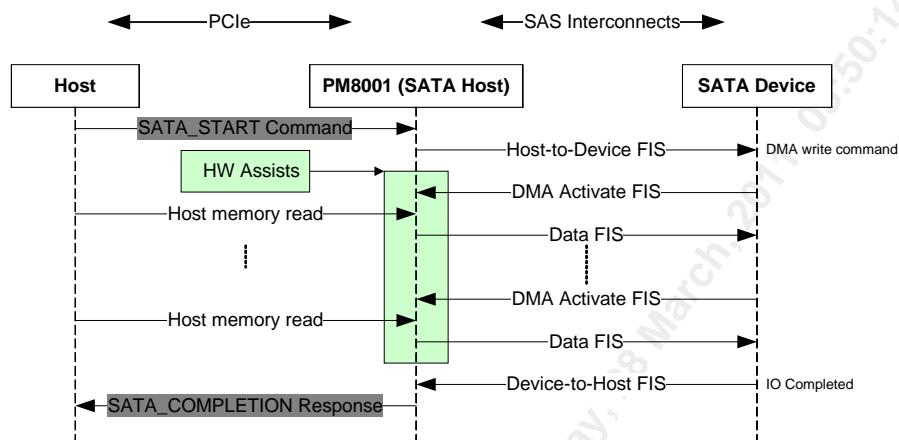
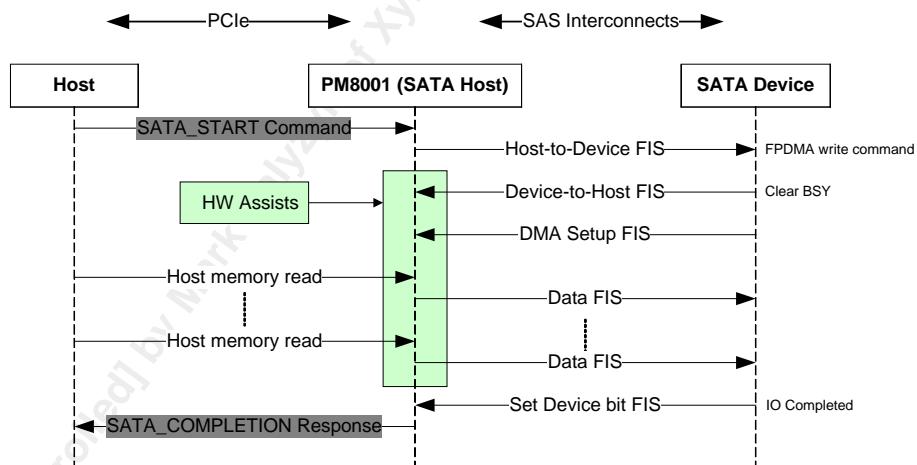
Figure 28 SATA Host Write Operation Flow Diagram (DMA Data Transfer Mode)


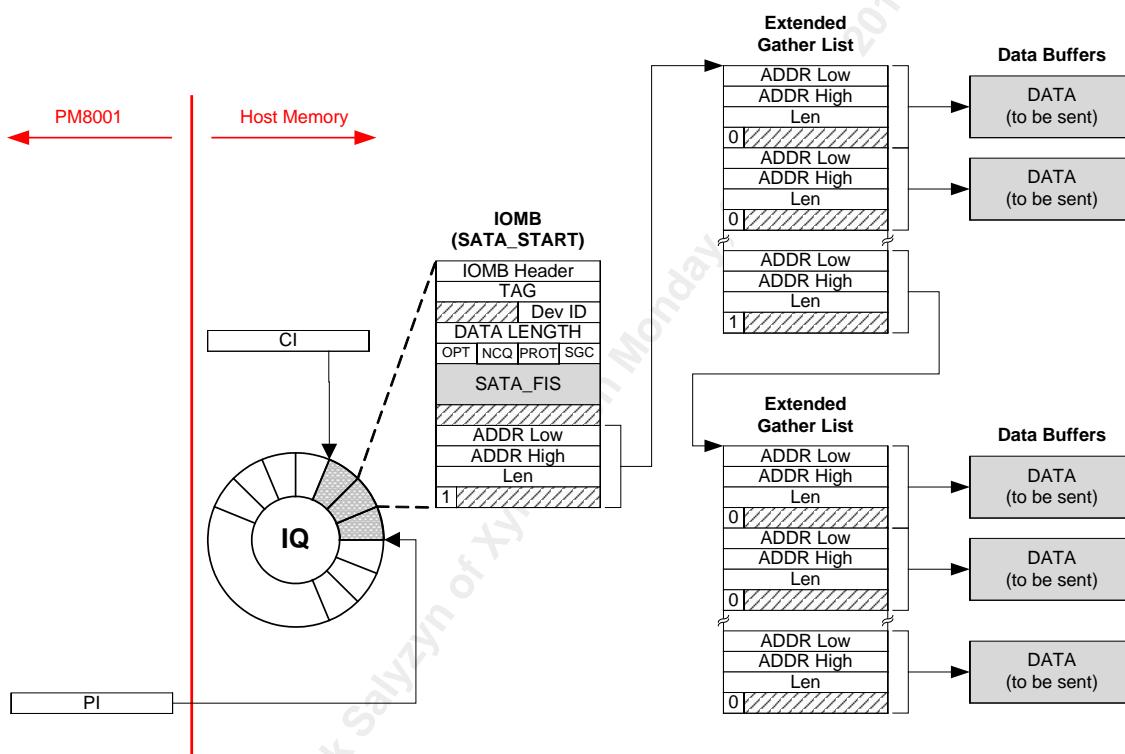
Figure 29 shows the interaction flow between a host, an SPC 8x6G controller acting as a SATA host, and a SATA device during a write operation using FPDMA data transfer mode.

Figure 29 SATA Host Write Operation Flow Diagram (FPDMA Data Transfer Mode)


3.13.1 SATA Host Write Using an Extended Gather List

[Figure 30](#) shows the data structures required to perform a SATA host write operation where data payload requires more than one data buffer segment.

Figure 30 SATA Host Write Example - Extended Gather List



To do a fully assisted SATA-HOST write operation to a SATA-DEVICE, the host selects the next available IOMB in the selected IQ. The host uses the [SATA_HOST_IO_START Command](#) to initiate a SATA-HOST I/O operation. (See Section 7.16, “[SATA_HOST_IO_START Command](#)”.)

The Extended (E) field of the [SATA_HOST_IO_START Command](#) must be set to 1b to indicate the use of an ESGL. A block of memory is then allocated as an extended gather list, the address of which is copied into the SGL Address fields: SGLAL and SGLAH.

The host then fills the addresses (SGLAL and SGLAH fields) and the corresponding lengths (LEN field) of the data buffers that describe the payload into the ESGL.

The host is free to define the length of each extended gather list. If more than one extended gather list is required, it can use multiple linked gather lists. To do this, the host sets the Extension bit (E) field in the last element of each gather list and provides the address of the chained gather list in the SGLAL and SGLAH fields. See Section 2.5, “[Scatter/Gather Lists \(SGLs\)](#)”.

The host indicates the type of SATA operation to perform in the ATAP field of the [SATA_HOST_IO_START Command](#). [Table 11](#) shows possible values of this field for a SATA read/write operation.

Table 11 SATA Read/Write Operation Modes

ATA PROT	Operation	Value
SATA_PROTOCOL_PIO	PIO protocol	0x05
SATA_PROTOCOL_DMA	DMA protocol	0x06
SATA_PROTOCOL_FPDMA	FPDMA protocol	0x07

The host indicates the direction of the data transfer, in this case a write operation, by writing 10b into the DIR field of the corresponding [SATA_HOST_IO_START Command](#). The destination SATA device is indicated in the DEVICE_ID field. See Section 3.2, “[Device Handle and DEVICE_ID](#)” for details. The number of data bytes to be transferred is specified in the DL field.

The host sets the HTAG field in the corresponding [SATA_HOST_IO_START Command](#) to a value that references the host I/O context. This field is returned in the command completion notification message.

If the SATA_PROTOCOL_FPDMA_WRITE transfer protocol is selected in the ATAP field, the host specifies the entry to use in the SATA device command queue in the NCQTAG field.

After the SATA Host to Device Frame Information Structure (FIS) is constructed in the SATAFIS field, the host instructs the SPC 8x6G controller to begin processing the I/O request. It does this by setting the Valid (V) bit in the IOMB header of the [SATA_HOST_IO_START Command](#) and incrementing the Producer Index (PI) of the selected IQ. After this, the SPC 8x6G controller owns the IOMB. See Section 6, “[Common IOMB Header](#)” for more details about the V bit.

The SPC 8x6G controller starts the SATA operation by sending the host-to-device FIS to the SATA-DEVICE. Depending on the ATAP field value in the [SATA_HOST_IO_START Command](#), the data transfer follows the PIO protocol (see [Figure 27](#)), the DMA protocol (see [Figure 28](#)), or the FPDMA protocol (see [Figure 29](#)).

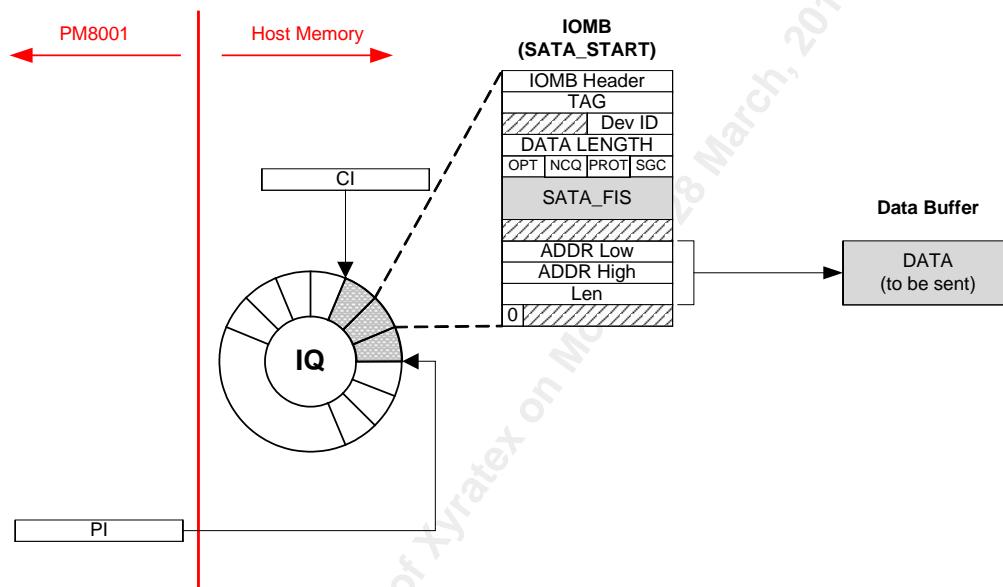
After fetching the IOMB containing the [SATA_HOST_IO_START Command](#) from host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section 2.3.2, “[IQ Producer and Consumer Indexes](#)”. At this point, the host owns the IOMB so it can be used for the next operation.

Once the SATA-DEVICE receives all of the data, it indicates the result of the transfer and the SPC 8x6G controller sends a [SATA_COMPLETION Response](#) message to the OQ designated in the OBID field of the inbound IOMB command header. See Section 6, “[Common IOMB Header](#)” and Section 8.9, “[SATA_COMPLETION Response](#)”. This signals the end of the SATA-HOST write operation. See Section 2.4, “[Outbound Queues \(OQs\)](#)”.

3.13.2 SATA Host Write Using a Local Gather List

[Figure 31](#) shows the data structures required to perform a SATA host write operation where the data payload is contained in a single buffer segment.

Figure 31 SATA Host Write Example - Local Gather List



A SATA-HOST write operation using a local data buffer descriptor operates in a similar manner to an extended gather list. In this case, a single data buffer descriptor is used to describe the host memory buffer containing the data to be sent. No external list structure is required. The host fills the address (SGLAL and SGLAH fields) and the length (LEN field) in the [SATA_HOST_IO_START Command](#) for the corresponding data buffer in host memory. (See Section 7.16, “[SATA_HOST_IO_START Command](#)” for details.)

3.14 SATA Host Read Operations

Figure 32 shows the interaction flow between a host, an SPC 8x6G controller acting as a SATA host, and a SATA device during a read operation using PIO data transfer mode.

Figure 32 Host Read Operation Flow Diagram (PIO Data Transfer Mode)

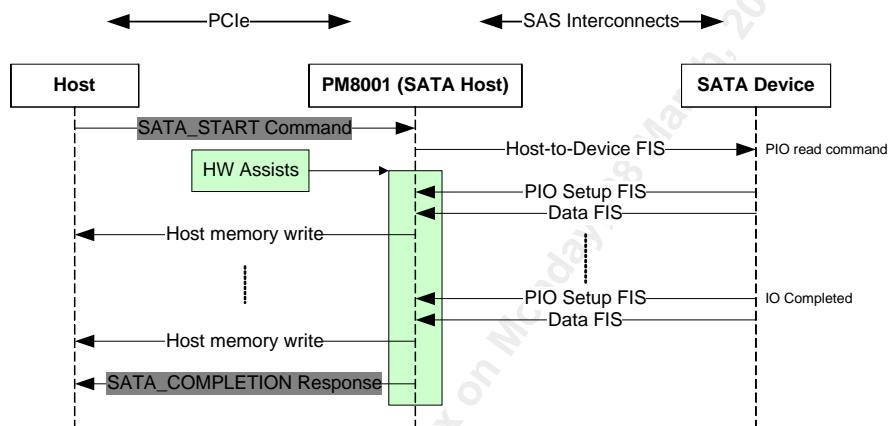


Figure 33 shows the interaction flow between a host, an SPC 8x6G controller acting as a SATA host, and a SATA device during a write operation using DMA data transfer mode.

Figure 33 SATA Host Write Operation Flow Diagram (DMA Data Transfer Mode)

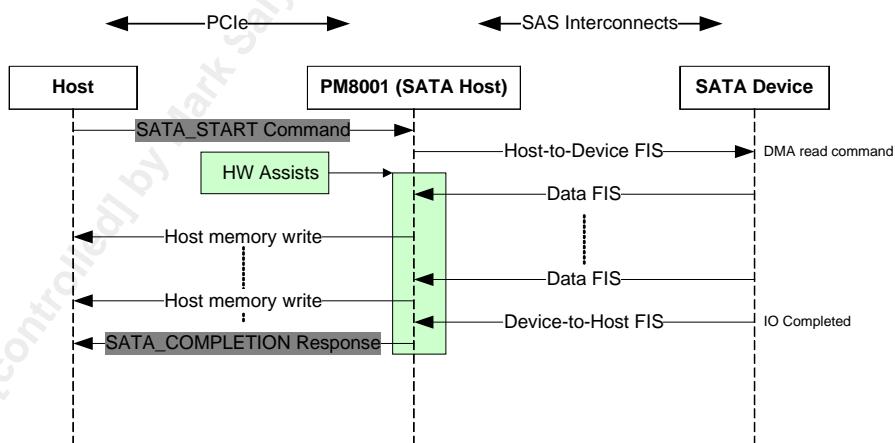
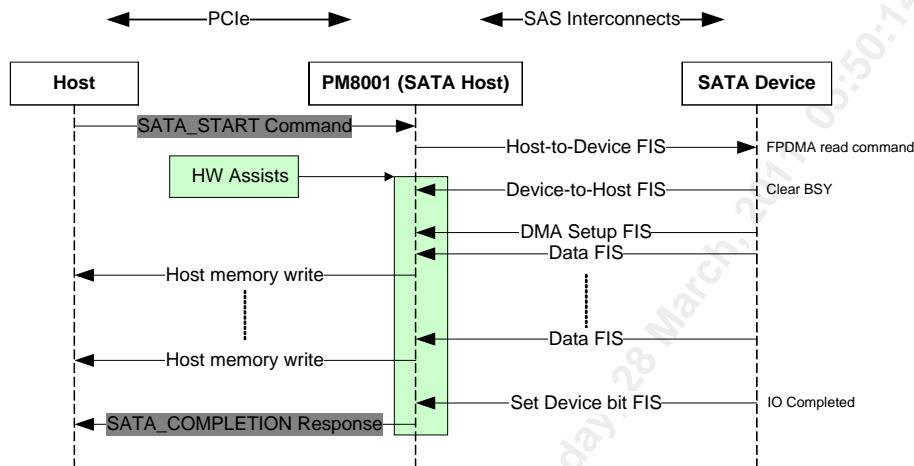
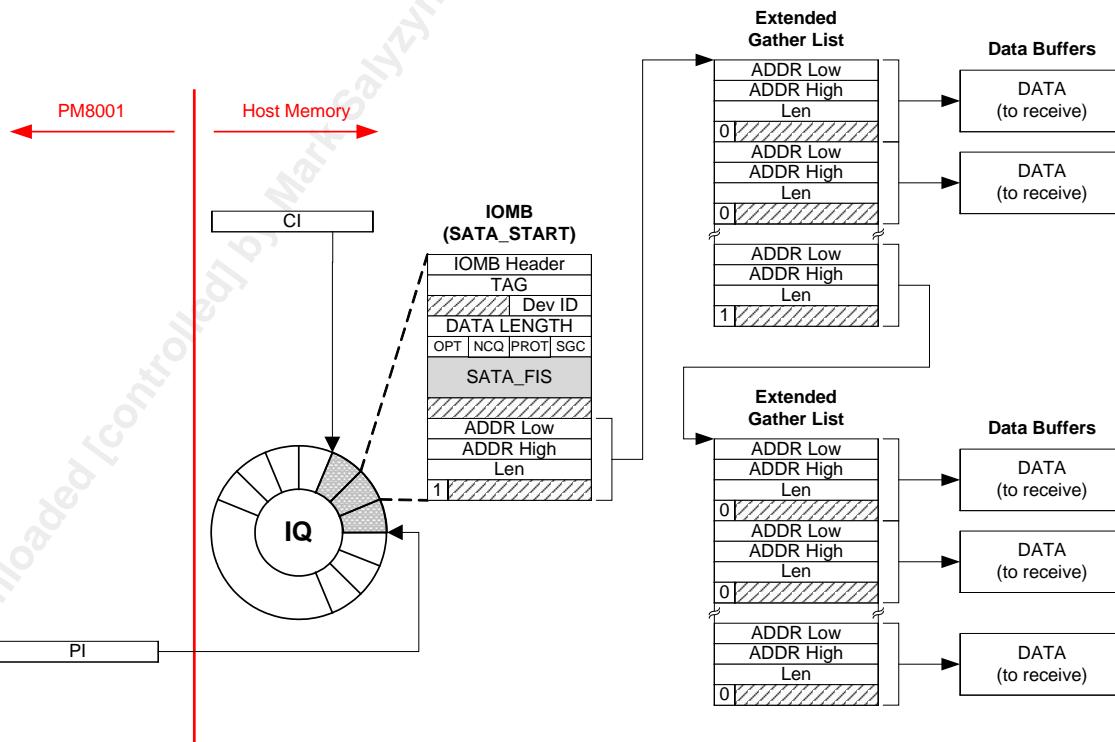


Figure 34 shows the interaction flow between a host, an SPC 8x6G controller acting as a SATA host and a SATA device during a read operation using FPDMA data transfer mode.

Figure 34 SATA Host Read Operation Flow Diagram (FPDMA Data Transfer Mode)


3.14.1 SATA Host Read Using an Extended Scatter List

Figure 35 shows the data structures required to perform a SATA host read operation where data to be received is stored in more than one data buffer segment.

Figure 35 SATA Host Read Example - Extended Gather List


To do a fully assisted SATA-HOST read operation from a SATA-DEVICE, the host selects the next available IOMB in the selected IQ. The host uses the [SATA_HOST_IO_START Command](#) to initiate a SATA-HOST I/O operation. (See Section 7.16, “[SATA_HOST_IO_START Command](#)”.)

The Extended (E) field of the [SATA_HOST_IO_START Command](#) must be set to 1b to indicate the usage of an ESGL. A block of memory is then allocated as an extended gather list, the address of which is copied into the SGL Address fields: SGLAL and SGLAH.

The host then fills the addresses (SGLAL and SGLAH fields) and corresponding lengths (LEN field) of the data buffers that describe the payload into the ESGL.

The host is free to define the length of each extended gather list. If more than one extended gather list is required, the host can use multiple linked gather lists. To do this, the host sets the Extension bit (E) field in the last element of each gather list and provides the address of the chained gather list in the SGLAL and SGLAH fields. See Section 2.5, “[Scatter/Gather Lists \(SGLs\)](#)”.

The host indicates the type of SATA operation to perform in the ATAP field of the [SATA_HOST_IO_START Command](#). Table 11 shows possible values of this field for a SATA read/write operation.

The host indicates the direction of the data transfer, in this case a read operation, by writing 00b into the DIR field of the corresponding [SATA_HOST_IO_START Command](#). The destination SATA device is indicated in the DEVICE_ID field. See Section 3.2, “[Device Handle and DEVICE_ID](#)”. The number of data bytes to be transferred is specified in the DL field.

The host sets the HTAG field in the corresponding [SATA_HOST_IO_START Command](#) to a value that references the host I/O context. This field is returned in the command completion notification message.

If the [SATA_PROTOCOL_FPDMA_READ](#) transfer protocol is selected in the ATAP field, the host specifies the entry to use in the SATA device command queue in the NCQTAG field.

After the SATA Host to Device FIS is constructed in the SATAFIS field, the host instructs the SPC 8x6G controller to begin processing the I/O request. It does this by setting the Valid (V) bit in the IOMB header of the [SATA_HOST_IO_START Command](#) and incrementing the Producer Index (PI) of the selected IQ. After this, the SPC 8x6G controller owns the IOMB. See Section 6, “[Common IOMB Header](#)” for more details about the V bit.

The SPC 8x6G controller starts the SATA operation by sending the host-to-device FIS to the SATA-DEVICE. Depending on the ATAP field value in the [SATA_HOST_IO_START Command](#), the data transfer follows the PIO protocol (see [Figure 32](#)), the DMA protocol (see [Figure 33](#)), or the FPDMA protocol (see [Figure 34](#)).

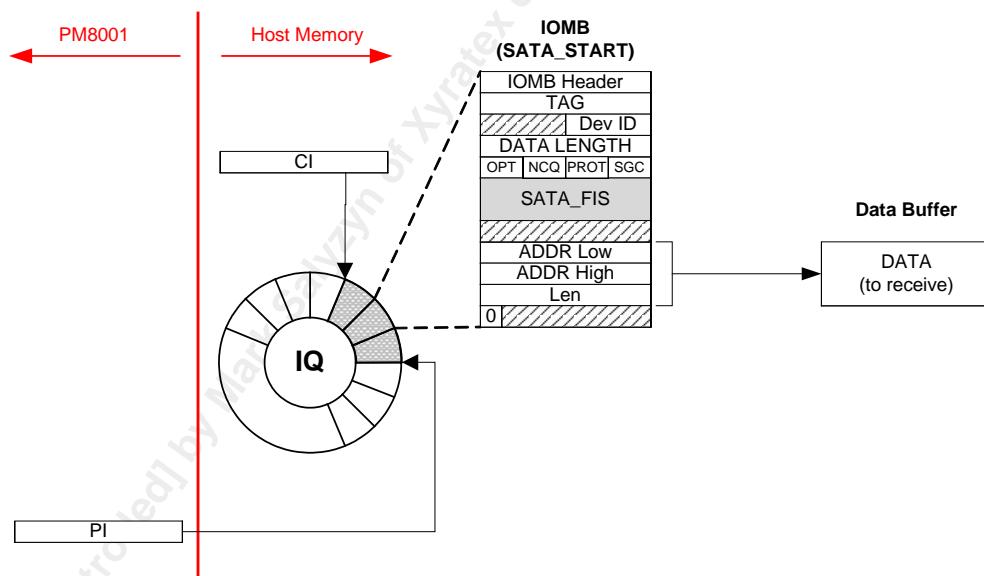
After fetching the IOMB containing the [SATA_HOST_IO_START Command](#) from host memory, the SPC 8x6G controller increases the Consumer index (CI) of the corresponding IQ. The SPC 8x6G does not clear the Valid (V) bit in the IOMB header. See Section [2.3.2, “IQ Producer and Consumer Indexes”](#). At this point, the host owns the IOMB so it can be used for the next operation.

Once the SATA-DEVICE transmits all of the data, it indicates the result of the transfer and the SPC 8x6G controller sends a [SATA_COMPLETION Response](#) message to the OQ designated in the OBID field of the inbound IOMDB command header. See Section [6, “Common IOMB Header”](#) and Section [8.9, “SATA_COMPLETION Response”](#). This signals the end of the SATA-HOST write operation. See Section [2.4, “Outbound Queues \(OQs\)”](#).

3.14.2 SATA Host Read Using a Local Scatter List

[Figure 36](#) shows the data structures required to perform a SATA host read operation where the data to be received is stored in a single buffer segment.

Figure 36 SATA Host Read Example - Local Gather List



A SATA-HOST read operation using a local data buffer descriptor operates in a similar manner to an extended gather list. In this case, a single data buffer descriptor is used to describe the host memory buffer where the received data is stored. No external list structure is required. The host fills the address (SGLAL and SGLAH fields) and the length (LEN field) in the [SATA_HOST_IO_START Command](#) for the corresponding data buffer in host memory. (See Section [7.16, “SATA_HOST_IO_START Command”](#).)

3.15 Firmware/Partition Update

The SPC 8x6G controller supports the update of firmware/partition images stored in attached flash memory. The host carries out the firmware/partition update by sending firmware update-specific IOMBs to the firmware running on SPC 8x6G. The firmware update mechanism is non-intrusive allowing other I/Os or commands to execute while an update is occurring in parallel. The same utility enables the updating of any partition in flash memory. The flash memory map illustrated in [Figure 44](#) in Section 4.5, shows the different partitions.

The following two IOMBs are used by the firmware update mechanism:

- [FW_FLASH_UPDATE Command](#) (Section 7.20)
- [FW_FLASH_UPDATE Response](#) (Section 8.15)

The firmware-upgrade mechanism:

- Supports 16-bit wide flash memory.
- Programs the image data in a piecemeal manner to support host systems that have limited memory for the image to be buffered before the flash memory programming starts.

See the firmware release notes [7] for information on which image files are used to update the firmware.

The header fields are shown in [Table 12](#).

Note: The image header is in Big Endian format.

The firmware image is segmented on the host into simple formatted blocks called flash packets. Segmenting the image allows the data path and the SPC 8x6G controller to transfer it efficiently, and reduces the system's buffering requirements.

The host must provide an offset field to accompany each packet. This offset is used as a check to ensure that data is being written to the flash memory is in the correct order. If any data is received out of order the download is aborted. The download can be restarted at any time by sending a flash packet with an offset of zero. For more information, see the [FW_FLASH_UPDATE Command](#) description in Section 7.20.

[Table 12](#) shows the image header fields in Big Endian byte order. This header only accompanies the first packet (indicated by an offset of zero). The image frame size is defined by the IOMB request sent by the host.

Table 12 Firmware/Partition Image Header

	Byte Offset	Data	Field Name	Field Description
Header	7:0	ASCII	Vendor ID	The vendor ID. The default values are 0x50, 0x4D, 0x43, 0x53, 0x0, 0x0, 0x0, and 0x0.

	Byte Offset	Data	Field Name	Field Description
	8	HEX	Product ID	The product ID. This should be set to 0x08.
	9	HEX	Hardware Rev	The hardware ID. This byte should be set to the hardware revision. The partition/firmware update algorithm checks this field against the hardware.
	10	HEX	Destination Partition	See Figure 44 and Table 35 in Section 4.5.
	11	N/A	Reserved	
	15:12	ASCII	Firmware Rev	This field is not used but contains the firmware revision of the code being downloaded.
	19:16	HEX	Image Length	<p>This field is used to check against the size of the destination partition.</p> <p>If the image length plus 8 bytes (for storing the CRC and the length at the end of a partition) is greater than the destination partition length then the status is reported as FLASH_UPDATE_HDR_ERR.</p> <p>If the total number of bytes received exceeds this number, is the status set to FLASH_UPDATE_LENGTH_ERR. The final CRC check is not done until this number of bytes is received.</p>
Firmware Data	23:20	HEX	Image CRC32	<p>This field is used to check the CRC of the data written to flash. The CRC is calculated using the same number of bytes as the specified image length.</p> <p>If the CRC check fails, the status is reported as a FLASH_UPDATE_CRC_ERR.</p>
	27:24	HEX	Startup Routine Address	This field is not used, but contains the value 0xBFC00000.
	<i>n</i> :28	HEX	Application Firmware Data	This field contains the application data as a binary image.

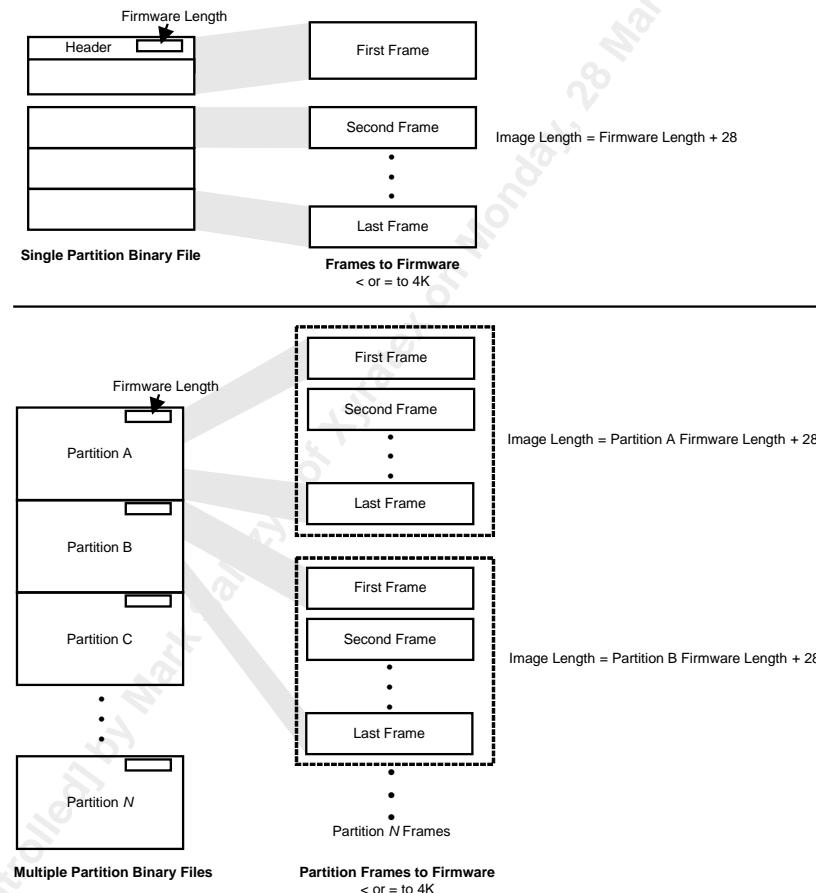
After each packet has been processed, the SPC 8x6G firmware sends the status of the firmware/partition upgrade process to the host through [FW_FLASH_UPDATE Response](#) IOMB. The host must wait for the [FW_FLASH_UPDATE Response](#) IOMB before sending another [FW_FLASH_UPDATE Command](#) IOMB.

On receipt of the last packet, the final CRC is computed and this CRC is then compared with the stored value from the firmware packet header. If they match, the firmware length and the CRC are written to the last 8 bytes of the partition. At this point a status of **FLASH_UPDATE_COMPLETE_PENDING_REBOOT** is returned and the device must be reset to re-invoke the bootloader image selection code.

Flash memory partition is described in Section 4.6, “[Flash Memory Partition Format](#)”.

The firmware binary files are provided in different formats. One format is a single binary image for a single partition in a single file and the other format contains multiple binary images for multiple partitions in a single file. For example an ila.bin file contains a single binary image whereas a fxxxxxx.x.bin file (x's are numbers) contain multiple binary images. The host application must ensure that in the multiple partition binary file, each partition image header is not mixed with previous partition data. [Figure 37](#) shows frame flow to the firmware. Note the total image length in the [FW_FLASH_UPDATE Command](#) refers to the current partition image length taken from the header and the number of header bytes.

Figure 37 Partition Binary Files – Frame Flow and Image Length Value



3.15.1 Flash Image Verification

Use the CRC-based image verification mechanism to allow the host and firmware to determine if a flash image is valid and integrated:

1. For a single partition image or binary file as shown in [Figure 37](#), the header of the image has the expected image CRC as described in byte offset [23:20] of [Table 12](#). Using the polynomial below, the host or firmware must calculate a new CRC and compare it to the value in offset [23:20] to verify the image integrity.

2. The host or firmware must use the CRC-32 IEEE 802.3 polynomial " $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1$ " to calculate the CRC of the image content. The length of the image used to calculate CRC is defined in byte offset [19:16] of [Table 12](#). This does not include the 28-byte header. The content of image begins right after the image header.

For a multiple partition image or binary files, as shown in [Figure 37](#), each partition has the same format as the single partition image. Repeat steps 1 and 2 above to check the CRC for each partition until reaching the end of the image file.

When the image is programmed successfully in flash memory, the last 32 bits of each partition are the image CRC, and the second last 32 bits in each partition are the length of the image in that partition.

3.15.2 Firmware Update Logic

The firmware update utility programs firmware images into the flash memory.

The SPC 8x6G evaluation board allows two firmware images to be programmed into flash memory: Image A and Image B. Typically in a board, Image A is programmed as the valid image and Image B is left blank. The SPC 8x6G will load the firmware from Image A. Image A also has the firmware upgrade software for image updates.

When host updates a new firmware image, firmware upgrades this to the backup image and on reset, the SPC 8x6G device boots from the latest image.

Note

- The initialization string (ISTR) image must be updated in a specific order to make sure the backup or primary image is updated correctly. When updating the backup image, a successful download of the IOP or AAP1 image must be completed, then the ISTR in the backup partition can be updated. If the IOP or AAP1 image update fails, then the ISTR image will be applied to the primary or active partition. This may result in the active firmware images having incompatible settings. To update the primary or active ISTR just follow the normal process for updating a firmware image.

When the image is upgraded:

1. The firmware update utility copies the new flash image to the inactive or backup image.
2. After the image is successfully loaded into the flash memory, the utility marks it as the active image.
3. On the next device boot, the SPC 8x6G's image loader application (ILA) loads the most recent active image and keeps the previous active image to the backup image. If the newly updated image is the same as the previously active image, both the previous active and backup images are kept.
4. The ILA checks the CRC-32 of the active image. If the CRC-32 shows the active image is valid, the ILA will boot from it. If the CRC-32 shows the active image is invalid, the ILA will check to see if the backup image is still valid:

- If the backup image is valid, the ILA will boot from it.
- If the backup image is also invalid, which should not occur in production firmware, the boot ROM will not load the firmware. Instead, an external software tool can be used to reprogram the flash memory.

3.16 GPIO Operation

The SPC 8x6G device provides up to 20 GPIO signals. The first 12 signals [11:0] are for customer use. The next 8 signals [19:12] are reserved for SPC 8x6G firmware. When the host performs GPIO setup or a read/write operation using the [GPIO Command](#), the host needs to make sure that it does not disturb the GPIO configuration for the [19:12] signals as these are reserved for SPC 8x6G firmware. (See Section [7.21, “GPIO Command”](#) for more details about this IOMB.)

Each signal can be configured either as an input or as an output. When configured as an output, the host can use the [GPIO Command](#) to set the desired level. GPIO inputs can also be configured so that the SPC 8x6G sends [GPIO_EVENT Notification](#) messages when specific GPIO events occur. (See Section [8.17, “GPIO_EVENT Notification”](#) for more details.)

After reset, the output type is set to 00b: tristated, input is enabled and no GPIO event is enabled. If a different configuration is required, the host must send the [GPIO Command](#) with the GS and GE bit set and the corresponding configuration fields set with the desired values.

Each GPIO input signal can be configured to generate [GPIO_EVENT Notification](#) messages to occur when an edge transition occurs (rising or falling).

[Table 13](#) shows the execution order when more than one operation is specified in the [GPIO Command](#).

Table 13 GPIO Operation Precedence

Operation	GPIO Command Field	Order
GPIO Signal Setup	GS	First
GPIO Event Setup	GE	
GPIO Write	GW	↓
GPIO Read	GR	Last

As indicated above, GPIO signals [19:12] are reserved for SPC 8x6G firmware. By default, the SPC 8x6G firmware assumes a driven configuration (which requires external pull-up resistors) and uses the following GPIO signals to control the LEDs:

Table 14 GPIO LED Control

GPIO	Description
GPIO-16	The LED is ON if any of SAS PHYs 0–3 (first quad connector) are up. The LED is OFF if all of the 0–3 SAS PHYs are down.

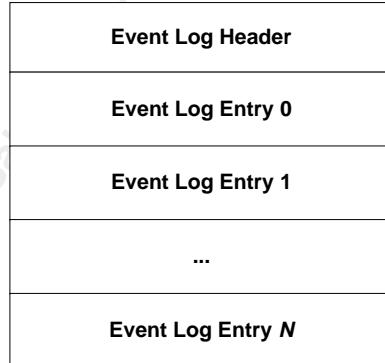
GPIO	Description
GPIO-17	The LED is ON if any of SAS PHYs 4–7 (second quad connector) are up. The LED is OFF if all of the 4–7 SAS PHYs are down.
GPIO-18	The LED is blinking (heartbeat) to indicate the SPC 8x6G MSGU AAP1 is active.
GPIO-19	The LED is blinking (heartbeat) to indicate SPC 8x6G IOP is active.

3.17 Event Log Operation

The SPC 8x6G firmware optionally supports the logging of firmware events to the host-allocated buffers. Two independent host buffers are used to log events specific to each processor (AAP1 and IOP). See Section 5.2, “MPI Configuration Table” for details on enabling this option and to set desired log severity level. To ensure all logs are in the host memory, read the “Event Log Latest Index”, wait for 50 ms, re-read the “Event Log Latest Index”. See Table 15 for a description of the “Event Log Latest Index”. If it is the same, the logs are all dumped from SPC 8x6G memory to host memory.

Figure 38 shows the format of the event log buffer.

Figure 38 Event Log Buffer Format



The event log buffer contains header information at offset 0x0 from the event log data buffer, followed by a series of fixed-sized event long entries starting from offset 0x20. The event log entries are encoded in a firmware-defined format. After the event log is retrieved, it is fed to an event log parser, which translates the event log entries to useful descriptions that can be displayed in a user-friendly way.

3.17.1 Event Log Header

The event log header is 32 bytes in size and the fields are described in the table below. The header size is not necessarily fixed in order to allow for future expansion.

Table 15 Event Log Header Format

DWord Offset	Field	Length	Description
0x00	Event Log Signature	4 Bytes	32-bit signature identifying the event log header. AAP1: 1234AAAA IOP: 5678CCCC
0x01	Event Log Entry Start Offset	4 Bytes	Specifies the byte offset of the first event log entry. The offset is calculated from the base of the event log buffer; that is, offset 0 of the event log header.
0x02	Reserved	4 Bytes	—
0x03	Event Log Buffer Size	4 Bytes	Size of the event log buffer in bytes, less the header size (32 bytes). Whereas the parameter set by the HOST in the MPI configuration table is the total size including the header.
0x04	Reserved	4 Bytes	—
0x05	Event Log Oldest Index	4 Bytes	Index pointer to the oldest entry in the log. Index is multiple of Event Log Entry Size.
0x06	Event Log Latest Index	4 Bytes	Index pointer to the latest entry in the log. Index is multiple of Event Log Entry Size.
0x07	Event Log Entry Size	4 Bytes	Size of each event log entry in bytes.

3.17.2 Event Log Entry

Event log entries are each 32 bytes (eight 32-bit words) in the following format.

Table 16 Event Log Entry Format

DWord Offset	Field	Length	Description
0x00	[2:0]: Number of Words in Log Entry [27:3]: Reserved [31:28]: Event Severity	4 Bytes	Number of valid words in the entry and the event severity level.
0x01	TimeStamp Upper	4 Bytes	The upper 32 bits of the time stamp. The timer resolution is in 8-nanosecond intervals. The timer is common to both the AAP1 and the IOP.
0x02	TimeStamp Lower	4 Bytes	The lower 32 bits of the time stamp. The timer resolution is in 8-nanosecond intervals. The timer is common to both the AAP1 and the IOP.
0x03	Sequence Number	4 Bytes	Unique sequence number that is common to both the AAP1 and the IOP. Used to identify chronological order of events across processors.
0x04	Log Word 0	4 Bytes	Log specific data.
0x05	Log Word 1	4 Bytes	Log specific data.

DWord Offset	Field	Length	Description
0x06	Log Word 2	4 Bytes	Log specific data.
0x07	Log Word 3	4 Bytes	Log specific data.

3.18 High Priority Operation and Normal Priority IOMB Processing

The SPC 8x6G supports normal and high priority queue processing.

A high priority request is performed by sending the IOMB to the inbound queues that have been configured as high priority. In addition, the H bit (high priority flag) in the IOMB header needs to be set. See Section 6, “Common IOMB Header” for the definition of IOMB header.

Figure 39 Messaging Unit Queue Processing

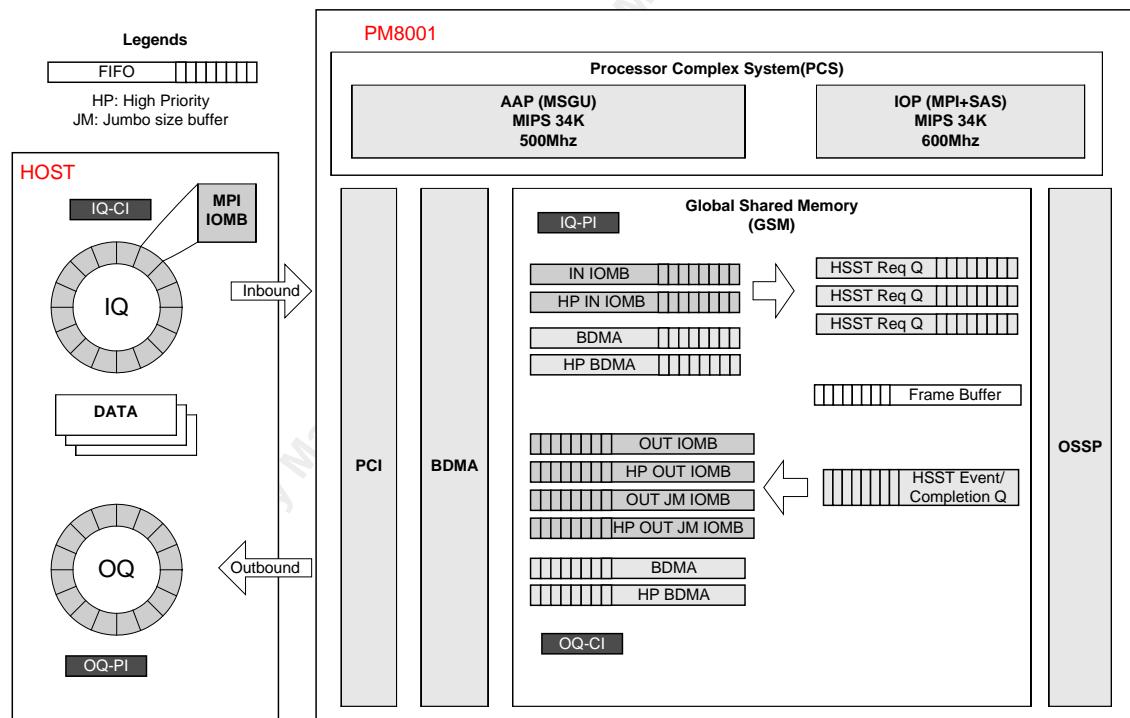


Figure 39 above shows some of the SPC 8x6G internal resources involved in the IOMB processing from the host IQ to the SPC 8x6G Messaging Unit (MSGU) and to the IOP, which translates inbound IOMB requests to OSSP/HSST Request Queue entries for transmitting the request to SAS/SATA link.

[Figure 39](#) also shows the outbound direction upon receipt of a SAS/SATA completion in the HSST Event/Completion queue, to the SPC 8x6G allocation of internal outbound IOMB resources and delivery to the host OQ.

The IOMB is moved between host memory and the SPC 8x6G GSM by means of a hardware BDMA engine. The BDMA operation for inbound operation is independent of the outbound direction. That is, concurrent BDMA operation for both inbound and outbound operation is supported.

The following subsections describe inbound and outbound IOMB processing.

3.18.1 Inbound Processing

The IOMB from the host IQ is fetched or moved to the SPC 8x6G's internal GSM by means of a hardware BDMA engine. The SPC 8x6G BDMA engine for the inbound direction supports both normal and high priority.

The SPC 8x6G MSGU initiates BDMA operation by writing the address of the BDMA descriptor to a FIFO register. As shown in [Figure 39](#), normal and high priority BDMA FIFO sets are used.

The BDMA descriptor includes the source and destination of the BDMA operation. For the inbound direction, the source address is the address of the inbound IOMB in host memory and the destination address is the internal SPC 8x6G GSM. Two internal SPC 8x6G GSM inbound pools are managed using two different FIFO sets: one for the normal-priority inbound IOMB (IN IOMB in [Figure 39](#)) and the other for the high-priority inbound IOMB (HP IN IOMB in [Figure 39](#)).

For normal priority, the IOMB is moved from a normal priority IQ in host to a normal priority IN IOMB pool in the GSM using the normal priority BDMA operation. For high priority, the IOMB is moved from a high priority IQ in the host to a high priority HP IN IOMB pool in the GSM using a high priority BDMA operation.

From the BDMA hardware perspective, all BDMA requests posted on the high priority inbound BDMA FIFO will be processed until it is empty, before the entries in the normal priority inbound BDMA FIFO are processed.

The SPC 8x6G firmware manages the individual IOMB fetching from host IQs. If more than one IQ is configured as a high priority queue, the SPC 8x6G firmware will fetch n entries from each high priority IQ in a round-robin fashion. The number of entries fetched from each individual high priority IQ is configured through the MPI Configuration Table. See the field, Inbound Queue High Priority Processing Depth (IQHPPD) in DWord 9 of [Table 38](#), “[MPI Configuration Table – Main Part](#)”.

Normal-priority IOMBs from normal-priority IQs will only be fetched until all of the high-priority IOMBs from all the high-priority IQs are fetched. If more than one IQ is configured as a normal priority queue, the SPC 8x6G firmware will fetch n entries from each normal priority IQ in a round-robin fashion. The number of entries fetched from each individual normal priority IQ is configured through the MPI Configuration Table. See the Inbound Queue Normal Priority Processing Depth (IQNPPD) field in DWord 9 of [Table 38, “MPI Configuration Table – Main Part”](#).

Once the inbound IOMB is fetched by the SPC 8x6G MSGU to the internal GSM, the SPC 8x6G IOP will continue processing the IOMB. The SPC 8x6G IOP will fetch entries from the HP IN IOMB FIFO until it is empty before fetching the normal priority IN IOMB FIFO.

During the processing of the inbound IOMB, the SPC 8x6G IOP allocates and posts an HSST Request Queue entry to the appropriate HSST Request queue based on the PORT_ID. From there, the HSST/OSSP hardware takes control of the SAS/SATA transaction.

3.18.2 Outbound Processing

At the completion of a SAS/SATA operation or the reception of a SAS/SATA link event, the SPC 8x6G IOP is notified with an entry in the HSST Event/Completion Queue describing the event or status of SAS/SATA operation completion.

Although there is no concept of dedicated priority for a host Outbound Queue, *an inbound high priority IOMB will always result in a high priority outbound path.*

Following the processing of an entry in the HSST Event/Completion Queue, the SPC 8x6G IOP prepares to pass the status/event information to the host in the form of an outbound IOMB to the host OQ.

The SPC 8x6G IOP allocates the internal GSM outbound IOMB resource from one of the several pools. There are separate GSM internal pools for normal priority and high priority outbound IOMBs. In order to support BC > 1 (see Section 6, “[Common IOMB Header](#)”), there are also jumbo (JM) size IOMB pools for normal priority and for high priority.

Just like the case for the GSM’s outbound IOMB pool separation based on priority, the SPC 8x6G BDMA engine for the outbound direction also supports normal and high priority.

For normal priority, the IOMB is moved from a normal priority IOMB in the GSM to the OQ in the host using normal priority BDMA operation. For high priority, the IOMB is moved from a high priority IOMB in the GSM to the OQ in the host using high priority BDMA operation.

From the BDMA hardware perspective, all BDMA request posted on the high priority outbound BDMA FIFO will be processed until it is empty before entries in the normal priority outbound BDMA FIFO are processed.

The SPC 8x6G MSGU fetches the entry (programs the BDMA to move the IOMB to host OQ) from the internal high priority outbound IOMB FIFO until it is empty, before fetching the normal priority outbound IOMB FIFO. If there are entries available on both HP OUT IOMB FIFO and the HP OUT JM IOMB FIFO, the SPC 8x6G MSGU will fetch one entry from each FIFO in a round-robin fashion.

Normal priority IOMBs from normal priority outbound IOMB FIFOs will only be fetched until all high priority IOMBs from all high priority Outbound FIFOs are fetched. If there are entries available on both the OUT IOMB FIFO and the OUT JM IOMB FIFO, the SPC 8x6G MSGU will fetch one entry from each FIFO in a round-robin fashion.

3.19 Host Interrupt Usage Model

The SPC 8x6G supports the following host-direct control interrupt mechanisms:

- 32 MSI interrupt vectors
- 16 MSI-X interrupt vectors
- Legacy INT-X

Host-direct interrupt control is done through the PCIe registers summarized in [Table 17](#).

Table 17 Host Controlled Interrupt Registers

Register Name	Section	Access	High Level Description
Outbound Doorbell Register	10.2.3	Host RO, Local R/W	Each bit of the ODR is mapped (hardwired) to a MSI-X (or MSI) vector table index entry. The SPC 8x6G sets the bit(s) to trigger the corresponding host interrupt vector(s) and the host may read the bit(s) in order to clear them through the Outbound Doorbell Clear Register .
Outbound Doorbell Clear Register	10.2.4	Host R/W, Local R/W	The host sets the bit(s) to clear the corresponding bit(s) in Outbound Doorbell Register .
Outbound Doorbell Mask Register	10.2.17	Host R/W, Local R/W	The host sets the bit(s) to mask host interrupts that would otherwise be caused by the respective bit being set in the Outbound Doorbell Register .
Message Unit Outbound Doorbell Auto Clear Register	10.3.8	Host R/W, Local R/W	This register is usually set once during initialization. When set, a bit in this register will cause the corresponding bit in the Outbound Doorbell register to automatically clear.
Interrupt Coalescing Timer Register	10.3.15	Host R/W, Local R/W	This register is a 16-bit interrupt delay timer with a resolution of 1 µs.
Interrupt Coalescing Control Register	10.3.16	Host R/W, Local R/W	This register is a 32-bit register that corresponds to each of the 32 potential interrupts. When a bit in this register is set, the corresponding interrupt will be delayed until the timer expires.

3.19.1 Setting Up Outbound Queue and Interrupt Vector Mapping

When MSI or MSI-X interrupt modes are enabled, the parameter Outbound Queue *n* Interrupt Vector (OQIVn) as described in Section 5.2.4, “[MPI Outbound Queue Configuration Table Fields](#)”, configures which interrupt vector is used for OQ *n*. The OQIVn parameter is a zero-based relative interrupt vector assigned to this instance of the SPC 8x6G.

The zero-based OQIVn is mapped as bit position in the:

- [Outbound Doorbell Register](#)
- [Outbound Doorbell Clear Register](#)
- [Outbound Doorbell Mask Register](#)
- [Message Unit Outbound Doorbell Auto Clear Register](#)
- [Interrupt Coalescing Control Register](#)

3.19.2 Legacy INT-X Usage Model

For host legacy INT-X usage, the host sets the [Message Unit Outbound Doorbell Auto Clear Register](#) to logic 0. The sequence of the interrupt processing is:

1. The SPC 8x6G MSGU writes into the [Outbound Doorbell Register](#) (the bit set according to OQIVn) after posting an outbound IOMB and advancing the OQ Producer Index (PI).
2. The SPC 8x6G PCIe subsystem generates an interrupt to the host.
3. The host interrupt service routine (ISR) is called, reads the [Outbound Doorbell Register](#), and handles the events.
4. The host ISR writes into the [Outbound Doorbell Clear Register](#) to clear the [Outbound Doorbell Register](#) and the interrupt.
5. The SPC 8x6G PCIe subsystem hardware de-asserts or clears the interrupt.
6. The host ISR may optionally mask the interrupt by writing to the [Outbound Doorbell Mask Register](#) when the host Windows DPC or the bottom half of the routine in Linux is used. The interrupt may be re-enabled again at the end of the DPC/bottom half routine.
7. For multiple OQs, the host repeats the OQ PI/CI and processes accordingly.

3.19.3 MSI or MSI-X 1-to-1 Mapping of Interrupt Vectors to OQs

In this mode of operation, the host sets the [Message Unit Outbound Doorbell Auto Clear Register](#) to 0xFFFFFFFF. The sequence is:

1. The SPC 8x6G MSGU writes into the [Outbound Doorbell Register](#) (bit used according to OQIVn) after posting the outbound IOMB and advancing the OQ Producer Index (PI).

The setting in [Message Unit Outbound Doorbell Auto Clear Register](#) clears the [Outbound Doorbell Register](#) bits automatically.

2. The SPC 8x6G PCIe subsystem generates an interrupt to the host.
3. The host ISR is called and handles the events. The host handles the events per the MSI(-X)-vector.

This is the most optimal mode of interrupt operation as the host does not need to:

- Read the [Outbound Doorbell Register](#) and write the [Outbound Doorbell Clear Register](#).
- Go through all the OQ PIs and CIs.

3.19.4 MSI or MSI-X Interrupt Vector Shared by Multiple OQs

In this mode of operation, the host can set the [Message Unit Outbound Doorbell Auto Clear Register](#) for those interrupt vectors dedicated to specific OQs. The sequence (for shared vectors) is:

1. The SPC 8x6G MSGU writes into the [Outbound Doorbell Register](#) (the bit used according to OQIVn) after posting an outbound IOMB and advancing the OQ Producer Index (PI).
2. The SPC 8x6G PCIe subsystem generates an interrupt to the host.
3. The host ISR is called, reads the [Outbound Doorbell Register](#), and handles events.
4. The host needs to go through the corresponding OQ PIs and CIs and processes them accordingly.
5. The host ISR writes into the [Outbound Doorbell Clear Register](#) to clear the [Outbound Doorbell Register](#) bit(s).

3.19.5 Interrupt Coalescing/Delay

The SPC 8x6G supports both firmware-managed interrupt coalescing and hardware-assisted interrupt coalescing. The host may only use one of these approaches, but not both.

3.19.5.1 Hardware-Assisted Interrupt Coalescing

The hardware-assisted interrupt coalescing is controlled by [Interrupt Coalescing Control Register](#) and [Interrupt Coalescing Timer Register](#). The timer value is applicable globally to the chip. That is, the timer delay cannot be controlled on the per interrupt vector basis. The masking of the interrupt coalescing can be done on a per bit basis in the [Outbound Doorbell Register](#). (That is, masking on a per interrupt vector basis.)

Note: Interrupt coalescing will not work with the auto-clearing feature of the [Outbound Doorbell Register](#). Hence, bits with the auto-clear feature set should not have the corresponding bit set in the [Interrupt Coalescing Control Register](#).

3.19.5.2 Firmware-Managed Interrupt Coalescing

Firmware-managed interrupt coalescing is controlled on a per OQ and per-interrupt vector basis. The firmware-based interrupt coalescing or delay is controlled by count and/or time delay parameters specified in the [MPI Outbound Queue Configuration Table Fields](#) as described in Section [5.2.4](#).

The parameter, Outbound Queue *n* Interrupt Coalescing Timeout (OQICTn), describes the maximum time, in 10 microseconds, the interrupt assertion interval from the last interrupt assertion to the host to the next interrupt assertion to the host. When multiple OQs are sharing the same interrupt vector, this value should be set to the same value for all OQs that share the interrupt. The parameter, Outbound Queue *n* Interrupt Coalescing Count (OQICCn), describes the maximum number of unmasked OQ *n* interrupt events accumulated (coalesced) before the SPC 8x6G asserts the interrupt signal.

3.20 Transport Layer Retry (TLR) Handling

With TLR enabled, when the SPC 8x6G:

- Receives a NAK when sending a DATA_OUT for the XFER_RDY, it will first abort the XFER_RDY that got NAKeD, and retry the whole XFER_RDY request.
- Detects an ACK/NAK when sending DATA_OUT for the XFER_RDY, it will retry the DATA_OUT only if the target device enables the TLR on its end by setting the Retry Data Frame bit on every XFER_RDY frame.
- Receives an NAK when sending COMMAND frame, it resends the COMMAND frame.
- Detects an ACK/NAK timeout when sending COMMAND frame, it does not resend the frame. The host application must query the target for this I/O and implement the appropriate recovery procedure as per the SAS 1.1 or SAS 2.0 specifications.
- Receives an NAK when sending RESPONSE frame as a target, it resends the RESPONSE frame.
- Detects an ACK/NAK timeout when sending a RESPONSE frame as a target, it resends the RESPONSE frame.

3.20.1 SAS 1.1 Transport Layer Retry

The SPC 8x6G allows a retry upon receiving a NAK or ACK/NAK timeout when transmitting frames. In general, a retry can be classified into retry data frames and all other frames (commands, responses etc). For example, in SAS, a command frame should not be retried, while a data frame may be if a NAK is received. In SATA, a command FIS should be retried, but not a data FIS. For an ACK/NAK timeout, the SPC 8x6G has no idea where the frame has been delivered to the device. If a retry is set for an ACK/NAK timeout, the frame will be re-sent and the target may get two copies of the same frame (if the ACK is lost on its return), the SPC 8x6G provides flexibility for the handling various retry scenarios.

During the [REGISTER_DEVICE Command](#) (Section 7.15), the R-flag in DWord 3 is used to enable or disable the retry feature. For this device, the SAS-1 TLR is enabled or disabled for a retry of all I/Os.

3.20.2 SAS 2.0 Transport Layer Retry

[Table 18](#) lists the bits added into the SSP frame header for the Transport Layer Retry (TLR) mode for SAS-2.

Table 18 SAS-2 TLR Mode Encoding

TLR Encoding	Definition
00	Use the mode page to configure the target for retry capability.
01	Enable TLR on per-command basis. This overrides the mode page. The initiator must check the retry data frame to see if a command should be retried.
10	Disable TLR and override mode page settings.
11	Use the mode page to configure the target for retry capability.

The host application may use this feature by manipulating the TLR in a per command basis. In every SSP command, the host application sets a corresponding two-bit TLR flag. When the SPC 8x6G processes the command, it treats the TLR flag as described in [Table 19](#).

Table 19 SPC 8x6G Processing of TLR for SSP Command

TLR Encoding	Definition
00	The SPC 8x6G retries the frame according to the R flag setting during the REGISTER_DEVICE Command (Section 7.15). The TLR flag should be set to "00" when the host application is running in the SAS 1.1 compatibility mode.
01	The SPC 8x6G enables a retry as described in Section 3.20. This overrides the default R bit setting that is set for this device. When the SPC 8x6G receives a NAK when sending a DATA_OUT for the XFER_RDY that the target sends, it checks the Retry Data Frame (RDF) bit in the current XFER_RDY frame. If the RDF bit is set to logic 1, the SPC 8x6G enables TLR. This overrides the mode page setting, that is, overrides the current retry behavior. If RDF is set to logic 0, the SPC 8x6G does not retry, overriding the mode page setting.
10	The SPC 8x6G disables the TLR, overriding the R bit specified in the per device setting for this command.
11	The SPC 8x6G retries the frame according to the R flag setting during the REGISTER_DEVICE Command (Section 7.15). The TLR flag should be set to "00" when the host application is running in the SAS 1.1 compatibility mode.

3.21 Host Direct Access (HDA) Mode

Host Direct Access (HDA) is a mechanism that enables the SPC 8x6G firmware image to be downloaded into the SPC 8x6G GSM and restarted during host driver initialization.

HDA mode is set as follows:

1. Bring the SPC 8x6G boot ROM into HDA mode as described in Section 3.21.1. The boot ROM runs on the AAP1 CPU.
2. The host detects that the boot ROM is in HDA mode and is ready to accept the HDA command. See Section 3.21.2 for a description of how the HDA protocol enables communication with the host.
3. The host executes the boot ROM HDA command, which downloads the ILA image from the host to the SPC 8x6G GSM. See Section 3.21.4, “[Host and SPC 8x6G Initialization Sequence](#)” for details.
4. The boot ROM authenticates the ILA image. When the image is successfully authenticated, the boot ROM jumps to HDA ILA code, which is still running on the AAP1 CPU. See Section 3.21.4, “[Host and SPC 8x6G Initialization Sequence](#)” for details.
5. The HDA ILA code running on AAP1 will start processing to bring and execute AAP1 and the IOP image from the host to the SPC 8x6G GSM. This requires that the synchronization protocol defined between the host and the SPC-8x6G ILA is followed. See Section 3.21.3, “[HDA ILA Protocol – Command and Response/Status](#)” for a description of the registers used for the synchronization protocol and Section 3.21.4, “[Host and SPC 8x6G Initialization Sequence](#)” for the detailed sequence.
6. Once Step (5) completes, the host determines that the SPC 8x6G HDA mode firmware is operational and is in the ready state by reading the MSGU register. As described in Section 5.1, “[MPI Configuration Table Access](#)”, the firmware ready state indication is reported in bits [1:0] of the [Scratchpad 1 Register](#) on and the [Scratchpad 2 Register](#). (See Sections 10.2.6 and 10.2.7 for details about bits [1:0]).
7. The host continues with the normal initialization sequence as described in Section 5.2.6.1, “[Host-SPC 8x6G MPI Initialization](#)”.

3.21.1 Bringing the Boot ROM Into HDA Mode

In order to set the HDA operating mode, one of the following methods can be used to bring the boot ROM into HDA mode:

- By configuring the Bootstrap pins LBI_A[14:13] to 10b. See Section 4.2, “[SPC 8x6G Bootstrap Configuration](#)” for details. This option is used when no flash ROM exists.
- By configuring the EEPROM ‘Force HDA Mode’ bit to 1b. This option is used when flash ROM and the firmware image might exist, but HDA mode is desired.
- By a Soft Reset (HDA Mode). See Section 11.4.3, “[Soft Reset Recovery \(HDA Mode\)](#)” for details.

- By an invalid ILA image. In this case, a Soft Reset (HDA Mode) is required to allow the hosts to start the HDA firmware download initialization sequence. See Section 11.4.3, “Soft Reset Recovery (HDA Mode)” and Section 3.21.4 “Host and SPC 8x6G Initialization Sequence”.

3.21.2 Boot ROM HDA Protocol – Command and Response

The boot ROM HDA protocol uses the last 64 bytes of the GSM exposed by the boot ROM to the host for the command and response block: 32 bytes of command structure followed by 32 bytes of response structure. The command is at offset 0x000FFEC0 and the response is at offset 0x000FFEE0 in the GSM (its start address is 0x0040_0000). If access is via MEMBASE-IV as shown in [Table 6](#), the command offset is 0x0000FEC0, and the response offset is 0x0000FEE0.

The host initiates a command and the target completes the command with a response. The mechanism for “sending” commands and “receiving” responses relies on the GSM. The host cannot write to the response block area. The target cannot write to the command block area.

The command block format is defined in [Table 20](#).

Table 20 Boot ROM HDA Protocol Command Format

Offset	Byte 3	Byte 2	Byte 1	Byte 0
0				Command Parameter 0
4				Command Parameter 1
8				Command Parameter 2
12				Command Parameter 3
16				Command Parameter 4
20				Command Parameter 5
24				Command Parameter 6
28	C_PA	SEQ_ID		CMD_CODE

Notes

- The Command Parameter fields store the command-specific parameters.
- The C_PA field is the command pre-amble. This value must be 0xCB. If this value is 0x00, the host is not in HDA mode.
- The SEQ_ID field stores the command sequence ID. This value must be incremented for subsequent commands. It cannot be 0. Valid values for the SEQ_ID field are from 0x01 to 0xFF, wrapping back to 0x01 at 0xFF. The target will set the entire command packet to 0 after a reset.
- The CMD_CODE field stores the command to be executed by the target. This 4-byte word starting at offset 28 must be written LAST and as an atomic unit. The target (boot ROM) continuously polls this word for an increment in the sequence ID (SEQ_ID) to trigger execution of the next command. The target requires that the C_PA is valid and that the SEQ_ID parameter is incrementing before accepting the new command. Of course, the CMD_CODE must also be valid.

The response block format is defined in [Table 21](#).

Table 21 Boot ROM HDA Protocol Response Format

Offset	Byte 3	Byte 2	Byte 1	Byte 0
0				Response Parameter 0
4				Response Parameter 1
8				Response Parameter 2
12				Response Parameter 3
16				Response Parameter 4
20				Response Parameter 5
24				Response Parameter 6
28	R_PA	SEQ_ID		RSP_CODE

Notes

- The Response Parameter fields store the response-specific parameters.
- The R_PA field is the response pre-amble. This value must be 0xDB. If this value is 0x00, it means that the target is not in HDA mode. The target must initialize this field to 0xDB when it enters the HDA mode.
- The SEQ_ID field stores the response sequence ID. This value must match the corresponding command SEQ_ID. The valid values for the SEQ_ID are from 0x01 to 0xFF, wrapping back to 0x01 at 0xFF. The target will set SEQ_ID to 0 after a reset.
- The RSP_CODE field stores the response code to corresponding CMD_CODE. This 4-byte word starting at offset 28 must be written LAST and as an atomic unit! The host must continuously poll this word for a change in the sequence ID that matches the corresponding command. A matching sequence ID indicates that the command is complete and the response is valid to be read.

The target initializes the entire command and response blocks to 0 after a reset and prior to enabling the PCIe interface.

The target must support at least the IDLE state where R_PA = 0xDB. The SEQ_ID is set to 0 on initial entry. Otherwise, it must be the same as the host command that forces the target into this state. Similarly, RSP_CODE must be set to 0 on initial entry; otherwise, the response code must be set correspondingly to the host command that forces the target into this state.

The command code range must be between 0x0001 and 0x7FFF. The response code range must be between 0x8000 and 0xFFFF.

3.21.2.1 Boot ROM HDA Commands and Responses

[Table 22](#) lists the HDA commands.

Table 22 Boot ROM HDA Commands

Mnemonic:	HDAC_BUFI	Code: 0x0001
Parameter [6:0]:	Not used	
Description:	Get target buffer information	
Response:	HDAR_BUFI	
Mnemonic:	HDAC_EXEC	Code: 0x0002
Parameter 0:	Entry offset – this value must be a multiple of 4 bytes	
Parameter 1:	Partition length – length of the partition in bytes, starting at offset 0	
Parameter [6:2]:	Not used	
Description:	Execute code in the buffer at the relative offset (to the start of the buffer) stored in Parameter 0	
Response:	HDAR_BAD_IMG and other implementation dependent status	
Mnemonic:	HDAC_RESET	Code: 0x0003
Parameter [6:0]:	Not used	
Description:	Place the target back into the IDLE state	
Response:	HDAR_IDLE	

[Table 23](#) lists the HDA responses.

Table 23 Boot ROM HDA Responses

Mnemonic:	HDAR_BUFI	Code: 0x8001
Parameter 0:	Buffer offset from the start of the GSM	
Parameter 1:	Buffer size in bytes	
Parameter [6:2]:	Not used	
Description:	Return target buffer information	
Mnemonic:	HDAR_IDLE	Code: 0x8002
Parameter [6:0]:	Not used	
Description:	Returning the target back into the IDLE state	
Mnemonic:	HDAR_BAD_IMG	Code: 0x8003
Parameter [6:0]:	Not used	
Description:	A response to HDAC_EXEC when the image cannot be validated successfully	
Mnemonic:	HDAR_BAD_CMD	Code: 0x8004
Parameter [6:0]:	Not used	
Description:	A response to any unrecognized command	
Mnemonic:	HDAR_INTL_ERR	Code: 0x8005
Parameter [6:0]:	Not used	
Description:	A response to any internally detected error	
Mnemonic:	HDAR_EXEC	Code: 0x8006
Parameter [6:0]:	Not used	
Description:	A response to HDAC_EXEC when the image is validated successfully and about to be executed	

3.21.2.2 Boot ROM HDA State

Once the PCIe interface is initialized, the boot ROM executes the Host Direct Access handshaking state machine, as shown in [Figure 40](#).

Figure 40 Boot ROM HDA State Diagram

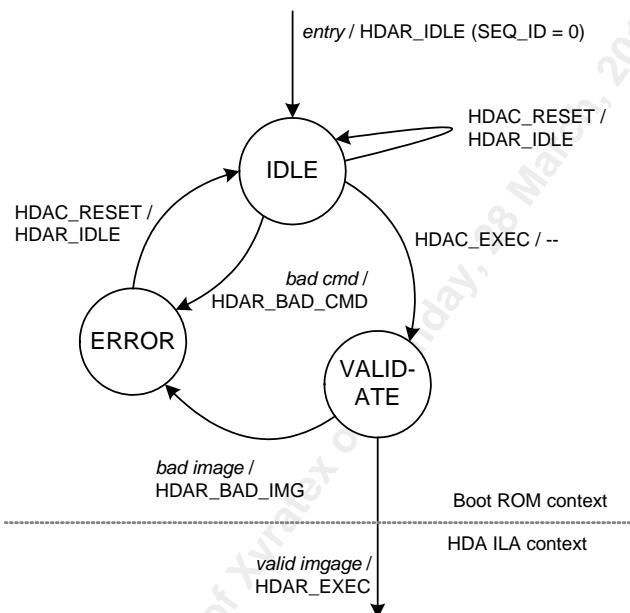


Table 26,
Step 2 of Boot ROM

While in the IDLE state, the host can query the buffer offset and size to determine where an ILA image can be written. For this purpose, the boot ROM always sets the buffer to start at offset 0 of the GSM so that the ILA image can be loaded. Furthermore, the size is set to a maximum of 1024 KB–64 bytes–256 bytes, where the 64 bytes is reserved for the HDA command/response and the 256 bytes is used for EJTAG.

Once the host issues the HDAC_EXEC command, the boot ROM begins validating the ILA image in the buffer. If successful, the boot ROM executes the image by jumping to the entry point specified in the HDAC_EXEC command in the context of AAP1. If the ILA image fails to be validated, the boot ROM enters the ERROR state and a HDAR_BAD_IMG is returned. The boot ROM stays in this state until the host sends a HDAC_RESET command.

The ILA image must be a raw, executable binary; neither ELF nor compression is permitted on the image. This image has to be properly signed with the appropriate CRC-32 signature.

The boot ROM HDA mode can be exited by a chip reset, or via a soft reset.

3.21.3 HDA ILA Protocol – Command and Response/Status

The host and the HDA-ILA command and response/status are communicated using a pair of registers:

1. The host writes the HDA-ILA command using the [Host Scratchpad 3 Register](#), described in Section [10.2.12](#).
2. The host reads the HDA-ILA response and status using the [Scratchpad 0 Register](#), described in Section [10.2.5](#).

The HDA-ILA command format in the [Host Scratchpad 3 Register](#) is as follows:

- The upper 8 bits [31:24] of the [Host Scratchpad 3 Register](#) describe the host command that the SPC 8x6G HDA-ILA firmware reads.
- The lower 24 bits [23:0] of the [Host Scratchpad 3 Register](#) describe the size of the image that host writes to the GSM.

Table 24 HDA-ILA Commands in the Host Scratchpad 3 Register

Mnemonic:	ILAHDAC_IOP_IMG_DONE	Code: 0x80
Lower 24 bits [23:0]:	The size of the image that the host writes to the GSM.	
Description:	Notifies the HDA-ILA that the IOP image has been copied into the GSM.	
HDA-ILA State:	ILAHDAA_IOP_IMG_GET	
Mnemonic:	ILAHDAC_AAP1_IMG_DONE	Code: 0x81
Lower 24 bits [23:0]:	The size of the image that host writes to the GSM.	
Description:	Notifies the HDA-ILA that the AAP1 image has been copied into the GSM.	
HDA-ILA State:	ILAHDAA_AAP1_IMG_GET	
Mnemonic:	ILAHDAC_ISTR_IMG_DONE	Code: 0x83
Lower 24 bits [23:0]:	The size of the image that the host writes to the GSM.	
Description:	Notifies the HDA-ILA that the initialization string image has been copied into the GSM.	
HDA-ILA State:	None	

The HDA-ILA response/status format in the [Scratchpad 0 Register](#) is as follows:

- The upper 8 bits [31:24] of the [Scratchpad 0 Register](#) describes the HDA status/state or response of the HDA-ILA firmware that the host driver reads.
- The lower 24 bits [23:0] of the [Scratchpad 0 Register](#) describes the offset in GSM where the host is to write the HDA firmware image.

Table 25 HDA-ILA Responses or State in the Scratchpad 0 Register

Mnemonic:	ILAHDAA_IOP_IMG_GET	Code: 0x10
Lower 24 bits [23:0]:	The offset in the GSM where the host is to write the HDA firmware image.	
Description:	The HDA-ILA is ready to receive the IOP image with the described offset.	
Mnemonic:	ILAHDAA_AAP1_IMG_GET	Code: 0x11
Lower 24 bits [23:0]:	The offset in the GSM where the host is to write the HDA firmware image.	
Description:	The HDA-ILA is ready to receive the AAP1 image with the described offset.	

3.21.4 Host and SPC 8x6G Initialization Sequence

This section describes the two-part host and SPC 8x6G initialization sequence: the host and boot ROM initialization followed by the host with HDA-ILA initialization. The images mentioned in the table are provided by PMC-Sierra in the form of a data array in the header files. Please consult the firmware release notes that are part of the release package available in the PM8001 PMC 8x6G content zone at <http://www.pmc-sierra.com/myPMC/> for further information on HDA-related files.

Notes

- To support 64-KB MEMBASE access, shifting the MEMBASE-III Inbound Window to the required location in the GSM is necessary. For details, see Section 2.6.1.
- The defaults in this procedure are in Big Endian format.

Table 26 Host and SPC 8x6G Initialization Sequence

Host		SPC 8x6G Boot ROM – First Part	
Step	Host Action	Step	Action
—	—	—	The boot ROM in HDA mode is running on AAP1. The boot ROM is polling for any HDA commands.
1	<p>The host reads the HDA response field RSP_CODE at byte offset 28:29 of the response block for HDAR_IDLE (0x8002). A value other than HDAR_IDLE (0x8002) indicates that the SPC 8x6G is not in HDA mode. Follow the steps described in Section 3.21.1 to bring SPC 8x6G into HDA mode.</p> <p>A reading of the correct RSP_CODE indicates that the SPC 8x6G boot ROM is ready to proceed to the next step for HDA initialization.</p>	—	—
2	<p>This step of writing the initialization string is needed in preparation for HDA-ILA execution since the initialization string is required at the beginning of HDA-ILA execution before the actual host and HDA-ILA command and response/state protocol is established.</p> <p>The host shifts the MEMBASE-III Inbound Window to the destination address 0x0047_0000, and copies the SPC 8x6G initialization string image to offset 0xE000 of MEMBASE-III.</p> <p>Note: The host must shift the MEMBASE-III window to the next 64 KB since the initialization string image starts at 0xE000, which is at 56 KB, and the initialization string image is larger than 8 KB.</p> <p>The way the host copies the image is</p>	—	—

Host		SPC 8x6G Boot ROM – First Part	
Step	Host Action	Step	Action
	<p>host/system-specific. Typically it is done by writing one DWord at a time to the correct offset.</p> <p>The host writes to the upper 8 bits [31:24] of the Host Scratchpad 3 Register described in Section 10.2.12, the command ILAHDAC_ISTR_IMG_DONE (0x83). The lower 24 bits [23:0] of the Host Scratchpad 3 Register indicates the size of the initialization string image that the host writes to the GSM.</p>		
3	The host writes the HDA Soft Reset SIGNATURE (0xA5AA27D7) to the Host described in the Host Scratchpad 0 Register Section 10.2.9. Writing the HDA Soft Reset SIGNATURE will prevent the HDA-ILA from reinitializing the PCIe block when the HDA-ILA is loaded.	—	—
4	The host shifts the MEMBASE-III Inbound Window to the destination address 0x0040_0000 (the GSM start address), and copies the SPC 8x6G HDA-ILA image to offset 0x0 of the MEMBASE-III. The way the host copies the image is host/system-specific, typically it is done by writing one DWord at a time to the correct offset.	—	—
5	<p>The host writes the HDAC_EXEC command (0x0002) via MEMBASE-IV for the HDA-ILA image for the boot ROM to authenticate and execute.</p> <p>The host sets parameter 0 and parameter 1 for the HDA-ILA image appropriately:</p> <ul style="list-style-type: none"> • Parameter 0: Entry offset – this value must be 0. • Parameter 1: Partition length – length of the partition in bytes, starting at offset 0. 	—	—
6	<p>The host continues polling for the HDA-ILA status via MEMBASE-IV. The polling timeout should be no more than 2 seconds.</p> <p>The response status, HDAR_EXEC, indicates a good response from the boot ROM. Other response states include HDAR_BAD_CMD or HDAR_BAD_IMG, which indicate a failure.</p>	1	The boot ROM executes the HDA_EXEC command.
		2	The boot ROM authenticates the HDA-ILA image. If it passes authentication, it sets the response status to HDAR_EXEC then jumps to the HDA-ILA code and executes the HDA-ILA code.
7	The host polls (reads) the upper 8 bits [31:24] of the Scratchpad 0 Register described in Section 10.2.5 for the ILAHDAAAP1_IMG_GET (0x11) state.	3	The HDA-ILA code is running on AAP1. It executes some initialization code and validates the initialization string that was passed by the host in host step (2) above.

Host		SPC 8x6G Boot ROM – First Part	
Step	Host Action	Step	Action
	<p>Polling timeout should be no more than 2 seconds. If polling timeout occurs, the host should check for a fatal error indication as described in Section 11.2 Device Specific Fatal Errors.</p> <p>Once the host detects that the HDA-ILA state is ILAHDA_AAP1_IMG_GET (0x11), it reads the lower 24 bits [23:0] of the Scratchpad 0 Register for the offset in the GSM (its starting address is 0x0040_0000) where the host is to write the HDA AAP1 firmware image.</p>		<p>If the initialization string validation is not successful, the HDA-ILA will report the Fatal Error with Scratchpad 1 Register bits [1:0], AAP_STATE, are set to 10b (error state), and Scratchpad 0 Register reports the details about fatal errors in the AAP1/MSGU.</p> <p>If the initialization string validation is successful, it then sets the upper 8 bits [31:24] of the Scratchpad 0 Register to the ILAHDA_AAP1_IMG_GET(0x11) state, and the lower 24 bits[23:0] to the offset in the GSM where the host is to write the HDA AAP1 firmware image.</p> <p>The HDA-ILA is polling for the HDA-ILA Command from the host. The next command expected is ILAHDAC_AAP1_IMG_DONE (0x81).</p>
8	<p>The host shifts the MEMBASE-III Inbound Window to the destination address = ((the offset in step (7) + 0x0040_0000) and 0xFFFF0000), i.e. 64 KB alignment address, and copies the first 64 KB (or less if the offset in step (7) is not 64 KB alignment address) of the AAP1 image to the offset (= (the offset in step (7) + 0x0040_0000) – the destination address) of MEMBASE-III. If the size of the AAP1 image is bigger than 64 KB or its content crosses the first 64 KB, you need to shift MEMBASE-III again to the next 64 KB, i.e., the destination address = ((the offset in step (7) + 0x0040_0000) and 0xFFFF0000 + 0x10000), and copy the next 64 KB (or less) of the AAP1 image to the offset 0x0 of MEMBASE-III, and so on until all of the AAP1 image is put into the GSM. The way the host copies the image is host/system specific, typically it is done by writing one DWord at a time to the correct offset.</p> <p>The host writes to the upper 8 bits [31:24] of Host Scratchpad 3 Register the command ILAHDAC_AAP1_IMG_DONE (0x81). The lower 24 bits [23:0] of the Host Scratchpad 3 Register indicates the size of the AAP1 image that the host writes to the GSM.</p>	—	
—	—	4	The HDA-ILA receives the command and validates the AAP1 image and copy (ELF-Copy) the image into the proper destination (AAP1 local memory).

Host		SPC 8x6G Boot ROM – First Part	
Step	Host Action	Step	Action
—		5	<p>The HDA-ILA sets the upper 8 bits [31:24] of the Scratchpad 0 Register to ILAHDAA_IOP_IMG_GET(0x10) state, and the lower 24 bits [23:0] to the offset in the GSM where the host is to write the HDA IOP firmware image.</p> <p>HDA-ILA is polling for the HDA-ILA Command from the host. The next command expected is ILAHDAC_IOP_IMG_DONE (0x80).</p>
9	<p>The host polls (reads) the upper 8 bits [31:24] of the Scratchpad 0 Register for ILAHDAA_IOP_IMG_GET (0x10) state.</p> <p>The polling timeout should be no more than 2 seconds. If polling timeout occurs, the host should check for a fatal error indication as described in Section 11.2 Device Specific Fatal Errors.</p> <p>Once the host detects that the HDA-ILA state is ILAHDAA_IOP_IMG_GET (0x10), the host reads the lower 24 bits [23:0] of the Scratchpad 0 Register for the offset in the GSM (its starting address is 0x0040_0000) where the host is to write the HDA IOP firmware image.</p>		
10	<p>The host shifts MEMBASE-III Inbound Window to the destination address = ((the offset in step (9) + 0x0040_0000) and 0xFFFF0000), i.e. 64 KB alignment address, and copies the first 64 KB (or less if the offset in step (9) is not 64 KB alignment address) of IOP image to the offset (= (the offset in step (9) + 0x0040_0000) – the destination address) of MEMBASE-III. If the size of IOP image is bigger than 64 KB or its content crosses the first 64 KB, you need to shift MEMBASE-III again to the next 64 KB, i.e., the destination address = ((the offset in step (9) + 0x0040_0000) and 0xFFFF0000 + 0x10000), and copy the next 64 KB (or less) of IOP image to the offset 0x0 of MEMBASE-III, and so on until all of the AAP1 image is put into the GSM. The way the host copies the image is host/system specific, typically it is done by writing one DWord at a time to the correct offset.</p> <p>The host writes to the upper 8 bits [31:24] of Host Scratchpad 3 Register the command ILAHDAC_IOP_IMG_DONE (0x80). The lower 24 bits [23:0] of the</p>	—	

Host		SPC 8x6G Boot ROM – First Part	
Step	Host Action	Step	Action
	Host Scratchpad 3 Register indicates the size of the IOP image that the host writes to the GSM.		
—	—	6	The HDA-ILA receives the command and validates IOP image and copy (ELF-Copy) the image into the proper destination (GSM).
11	The host writes 0x0 to the Host Scratchpad 0 Register .		
12	The host polls the Scratchpad 1 Register bits [1:0] AAP_STATE and the Scratchpad 2 Register bits [1:0] IOP_STATE to go to 11b (Ready state).	7	Continues with the normal AAP1 and IOP start-up sequence.
—	—	8	When the AAP1/MSGU initialization completes, the Scratchpad 1 Register bits [1:0] AAP_STATE to 11b (Ready state) are set. When the IOP MPI_APP initialization completes, the MSGU Scratchpad 2 Register bits [1:0] IOP_STATE to 11b (Ready state) are set.
13	The host continues with the normal SPC 8x6G Configuration Table initialization sequence as described in Section 5.2.6.1 Host-SPC 8x6G MPI Initialization .	—	—

3.22 SMP Request Serialization

This section describes the initiator mode SMP request control and queuing on the SPC 8x6G as well as the host control to configure the queuing and serialization of the SMP requests.

The SPC 8x6G internal queuing scheme achieves the following:

- Limits the concurrent SMP requests in a wide-port configuration in order to allow some of the PHYs to be used for normal I/O operation.
- Removes/reduces the probability of FIFO head-of-line blocking in the hardware HSST Request Queue (RQ) FIFO when multiple SMP requests for the same device are queued up (serially) in the HSST RQ FIFO.

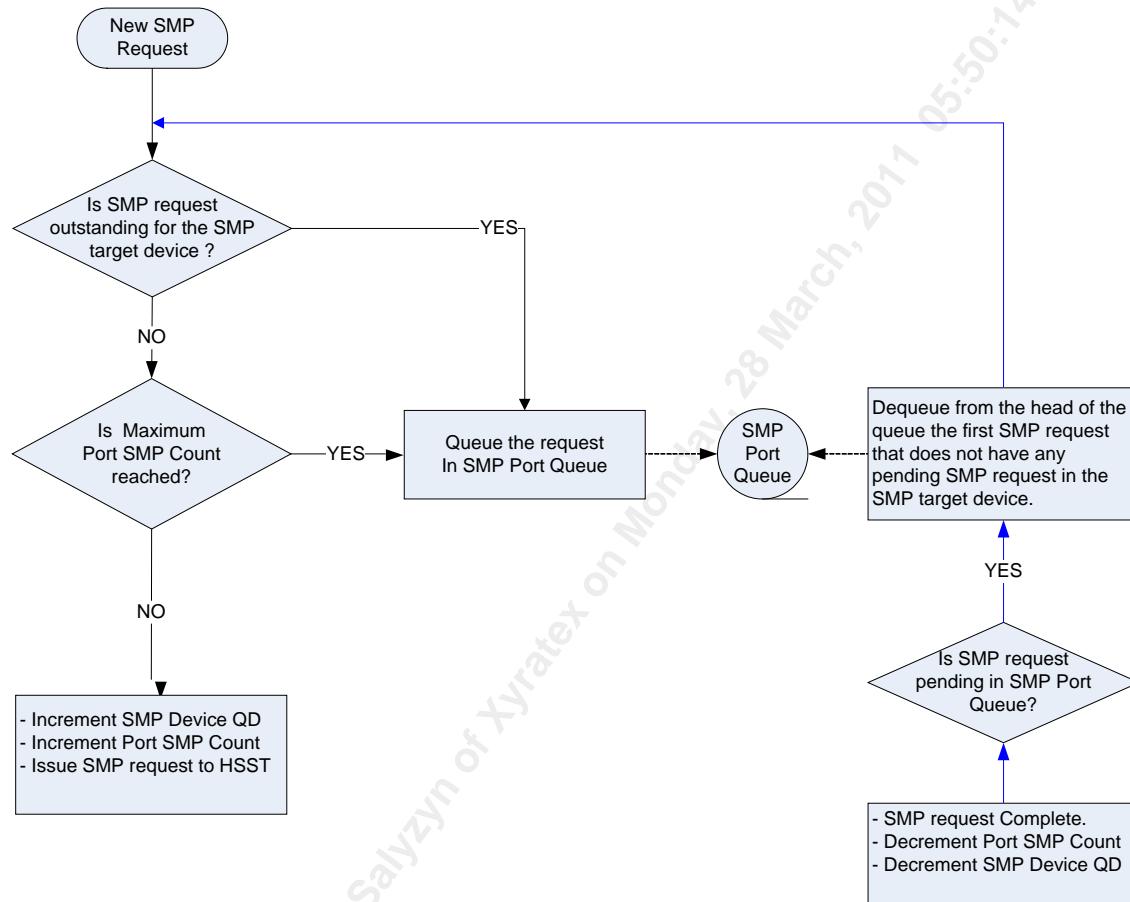
The SPC 8x6G firmware has a per-port eight-entries deep queue that allows the host to send two sequential SMP requests to the same SMP target device, that is, one will immediately be put into the HSST RQ and the other will be queued by the firmware. The firmware has a per SMP target device queue depth counter indicating how many SMP requests are pending on the SMP target. Since the firmware can only queue up 8 entries per controller, new SMP requests that the SPC 8x6G firmware receives that cannot be put into the queue will be returned to the host using an [SMP_COMPLETION Response](#) described in Section 8.4 with the status IO_ERROR_INTERNAL_SMP_RESOURCE.

The host must be aware of the number of PHYs in the wide port and manage how many pending SMP requests can be issued sequentially without waiting for SMP completion. This is done to limit the available PHYs used for SMP and leave some of the PHYs for normal I/O.

The IOMB [PORT_CONTROL Command](#) with the opcode set to SET_SMP_PHY_WIDTH is used to allow host to set the number of maximum concurrent SMP requests using the PHYs in a port. (See Section 7.26, “[PORT_CONTROL Command](#)” for details.)

[Figure 41](#) shows the SMP queuing process for both the inbound and outbound paths.

Figure 41 SMP Queuing and Serialization Diagram



Notes

1. The Maximum Port SMP Count records the maximum SMP requests that are pending on the port. This corresponds to the maximum number PHYs that can be used to send an SMP request. This number is set by the host by sending the IOMB PORT_CONTROL Command with the opcode set to SET_SMP_PHY_WIDTH.
2. The Port SMP Count is the current SMP requests that are pending on the port. This corresponds to the current number of PHYs that are used for an SMP request.
3. The SMP Device Queue Depth (QD) is the SPC 8x6G internal firmware count of pending SMP requests to a specific SMP target device.
4. The SMP Port Queue is the per port SMP queue with maximum queue depth of 8 entries where an SMP request that cannot be sent to the HSST is queued. The queue depth cannot exceed the number set by the [PORT_CONTROL Command](#) (with the opcode set to SET_SMP_PHY_WIDTH). If the number of SMP requests that need to be queued exceeds the maximum number set in the [PORT_CONTROL Command](#), the SPC 8x6G will return the SMP request to the host with the [SMP_COMPLETION Response](#) set with the status IO_ERROR_INTERNAL_SMP_RESOURCE.

For the inbound path:

- The incoming SMP request is queued in the SMP Port Queue if the SMP target device already has an outstanding request or the Port SMP Count has reached its maximum value. A check is made in that order.
- Otherwise, the SMP request is dispatched to the HSST RQ FIFO, after incrementing the SMP Device Queue Depth (QD) and Port SMP Count.

For the outbound path:

- The completed SMP request is posted to the host, decrementing the Port SMP Count as well as the SMP Device Queue Depth.
- If the SMP Port Queue has any requests, remove the first SMP request from the head of the queue destined for the SMP target device with no outstanding requests. No re-ordering will be done for the queue entries that remain in the queue.

The ordering of SMP requests sent on the wire to the SMP targets is, in most cases, dictated by the host ordering when sending the SMP request to the SPC 8x6G.

If there is only a single PHY in the port, the queuing scheme is still taking place.

3.23 Expansion ROM Support

The SPC 8x6G controller supports optional expansion ROM space. An UEFI-based driver can be stored in the 1-MB expansion ROM image location in flash memory. The system will use value of the PCI Configuration register, ROMBASE at offset 0x30 to access the expansion ROM space. See the expansion ROM image section of [Table 35](#) for details about the partition address.

Use the [GET_NVMD_DATA Command](#) to read back the expansion ROM image in flash memory and the [FW_FLASH_UPDATE Command](#) to write the new expansion ROM image to flash memory. Note: Any changes to the expansion ROM partition require a complete erase and re-write of that memory location. If there are multiple drivers in this location, if one driver requires updating all of the drivers will require rewriting in this flash partition.

3.24 Customizing Firmware Behavior

To provide additional flexibility and preserve backward compatibility, the SPC 8x6G provides additional flags in the MPI Configuration Table to determine the actual action. (See Section [5.2.1](#), Dword 0x13 CUSTSET)

3.24.1 IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY

The status of IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY 0x3F can occur for two reasons:

1. Lack of a Rx buffer for an SMP request for 100us.
2. I/Os are drained by hardware to prevent a potential lockup.

Bit #0 of the MPI Configuration Table DW #0x13 has an optional setting to provide a separate status for the second case:

0: Default. Both cases will be reported with the status of IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY.

1: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT 0x43, will be reported for I/Os drained by hardware as lockup prevention.

The host's action is only to retry the failed I/O while it may choose to get a different event.

3.24.2 IOP_EVENT_IT_NEXUS_LOSS

The host will not get an I/O completion with the status of IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS if it aborts the I/O before the IT nexus loss timer expires. Bit #1 of the MPI Configuration Table DW#0x13 is used to report an additional event with SAS_HW_EVENT notification:

0: Default. No additional event.

1: IOP_EVENT_IT_NEXUS_LOSS, 0x22, is reported when an I/O receives an IT Nexus Loss timeout. The EVPARAM field is the device ID (32 bits) where the event is received. The STATUS field is the status of the operation:

0x00: SAME. Device did not receive IT Nexus Loss timeout.

0x01: NON_OPERATIONAL. Device was already set to non-operational.

0x02: UNUSED. Device was deregistered.

0x03: REUSED. Device was set by a TM command to DS_IN_RECOVER or the same device ID (lower 16 bits) was deregistered and reassigned.

3.24.3 IOP_EVENT_PHY_LOCKUP_DATAOUT

When an STP connection was closed successfully prior to an SSP OPEN (Data phase) which failed (IT nexus loss timeout, etc.), a PHY in the SPC 8x6G may lock up. This issue is addressed by resetting the hardware at the PHY from which the stuck I/O is sent. The host has the option to receive a notification when the lockup condition occurs.

Bit #2 of the MPI Configuration Table DW#0x13 is used to report an additional event with SAS_HW_EVENT notification:

0: Default. No additional event is reported when the SPC 8x6G resolves the lockup.

1: IOP_EVENT_PHY_LOCKUP_DATAOUT, 0x23, is reported when the SPC 8x6G starts the resolution.

This new event will be reported before the SPC 8x6g resets the PHY.

3.24.4 IOP_EVENT_PHY_DOC_ABORT_TMO

The SPC may not be able to abort a write command in the dataout phase before the timeout expires. To complete the abort command, the SPC forces the I/O out of hardware by resetting the PHY so a SAS_HW_EVENT(IOP_EVENT_PHY_DOWN) is expected. The host can use this extra event to identify a PHY down situation caused by the resolution. Bit #3 of the MPI Configuration Table DW#0x13 is used to report the additional event with SAS_HW_EVENT Notification:

0: Default. No additional event is reported.

1: SAS_HW_EVENT(IOP_EVENT_PHY_DOC_ABORT_TMO, 0x24, is reported before the SPC 8x6G resets the PHY.

3.24.5 RETURN_ERROR_ON_OUT_OF_IOST_RESOURCE

By default, the IOP stopped processing IOMBs when it ran out of IOST entries. This blocked all IOMB processing even if the IOMB was not I/O related.

Bit #4 of the MPI Configuration Table DW#0x13 is used to control SPC FW behavior when encountering an out of IOST resource condition.

0: Keeps the default FW behavior.

1: Informs the SPC to return I/O related requests immediately with a status code (0x1004) when it runs out of IOST entries. This gives the firmware the chance to process subsequent IOMBs that do not require an IOST entry.

3.24.6 ENABLE_OUTBOUND_PROCESSING_FAIRNESS

By default, the SPC firmware remained in the outbound processing loop as long as there were pending IMQ entries. This may have caused the SPC to stall inbound processing when there were many IMQs to process.

Bit #5 of the MPI Configuration Table DW#0x13 is used to enable outbound processing fairness:

0: Keeps the default FW behavior.

1: Enables the SPC to process a maximum of 16 IMQ messages in the outbound loop before switching to inbound IOMB processing.

3.25 GPIO Operation

The SPC 8x6G supports two GPIO interfaces as required by the SFF 8485 Specification for Serial GPIO(GPIO) Bus. Each GPIO interface supports four PHYs. This method serializes general purpose I/O signals. The SFF 8485 standard defines communication between an initiator and the target. For more details see the SFF 8485 standard.

The standard defines GPIO registers that need to be read/write. The configuration register controls enabling/disabling of GPIO signals, blink generator rate, stretch activity ON/OFF and maximum activity ON/OFF for GPIO LEDs. There are three types of GPIO LEDs: Locate, Activity and Error. They are used to indicate different GPIO patterns.

The SPC 8x6G has default settings for GPIO. It is enabled and configured by setting a definitive pattern in the GPIO_CFG[0] register. The definitive pattern is defined in the initialization string of the SPC8x6G.

With this pattern, when the host sends a PHY_START command, the locate LED for that PHY turns ON when the PHY is in SAS_PHY_READY state and turns OFF when transitioning out of SAS_PHY_READY state. Similarly for I/O activity, the SPC8x6G transmits a blinking pattern on the activity LED to indicate I/O transfer is progressing. The activity LED shows the pattern for the duration of the activity.

This is a default SPC8x6G setting, the host can use different GPIO patterns based on the SFF 8485 specification. The host will be able to generate different patterns with the GPIO IOMB.

For details of the GPIO IOMB see Section [7.33](#).

4 Initialization and Configuration

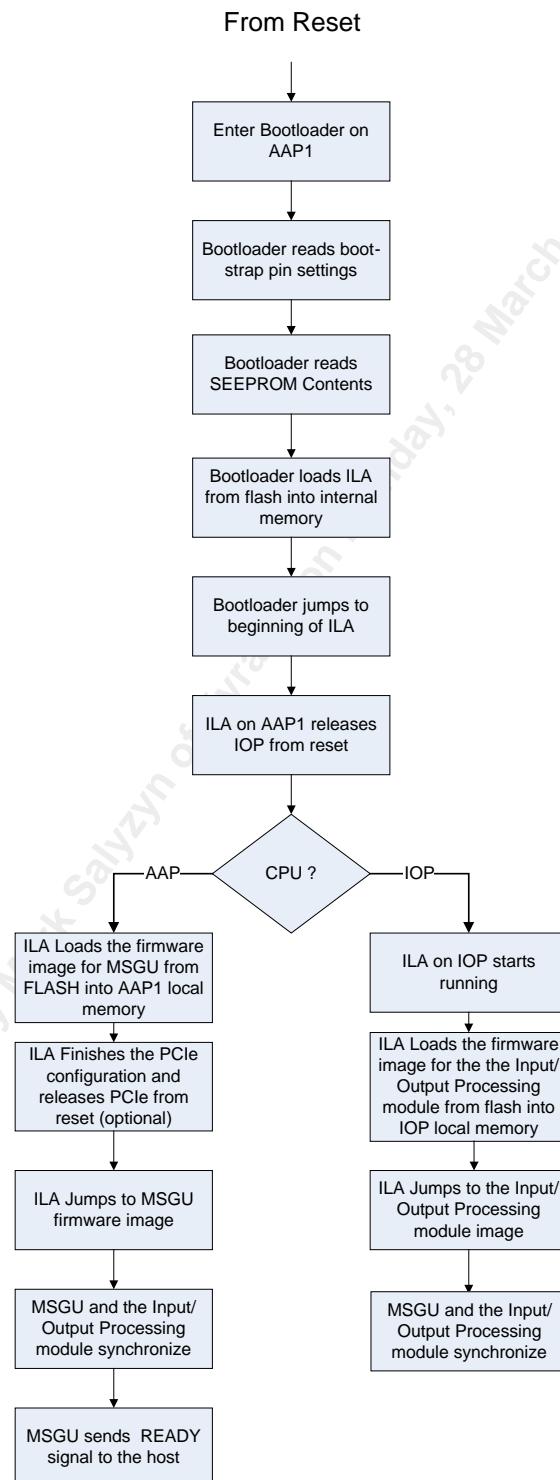
The SPC 8x6G bootloader and image loader agent (ILA) firmware components provide a means to load and verify firmware images into the SPC 8x6G controller for execution.

The bootloading process occurs in two steps:

1. After a power-up or a device-reset, the ROM-resident bootloader does the initial verification and loading of the ILA. The ILA code typically resides in an external flash memory device.
2. When executed, the ILA verifies the SPC 8x6G firmware image signature and, if the signature is valid, loads the firmware image to its designated processor.

[Figure 42](#) shows the initialization flow of the SPC 8x6G controller.

Figure 42 SPC 8x6G Initialization Flow



4.1 Power-up and Initialization

During power-up or a PCI soft reset, the SPC 8x6G boot code initializes the internal AAP1 and IOP, reads the bootstrap configuration, and executes as specified in the bootstrap configuration.

4.2 SPC 8x6G Bootstrap Configuration

The following table summarizes the functions of the SPC 8x6G bootstrap pins and their recommended configuration.

Table 27 Bootstrap Configuration

LBI_A[x]	Bootstrap Function	Description
0	Reserved	Needs to be set to 0
1	Reserved	—
9:2	TWI_ADDR_SEL[7:0]	External EEPROM device address.
10	LBI_WRD_BYTB	0: Local data bus is 8 bits wide. 1: Local data bus is 16 bits wide. The value of this field is ignored by the bootloader and ILA.
11	Reserved	Must be set to 0.
12	Reserved	Must be set to 0.
14:13	BOOT_SOURCE[1:0]	00: Do not boot 01: Boot from flash memory on LBI CE0 1x: Boot from PCIe (HDA mode)
15	Reserved	Must be set to 1.
16	Reserved	Must be set to 0.
17	Reserved	Must be set to 0.
18	Reserved	Must be set to 0.
21:19	Reserved	—
22	Reserved	Must be set to 1.
23	Boot EEPROM Size	0: EEPROM supports 16-bit addressing. 1: EEPROM supports 8-bit addressing (default).
24	Reserved	Must be set to 0.
25	Reserved	Must be set to 1.
26	Reserved	Must be set to 0.

4.3 Configuration SEEPROM

Additional configuration data can be provided to the SPC 8x6G bootloader using the configuration SEEPROM device connected to TWI 0.

This configuration SEEPROM is divided into two sections:

- General configuration (mandatory)
- Optional configuration

The mandatory general configuration section is 64 bytes, while the optional section varies in size depending on the number/type of optional tables appended.

The configuration SEEPROM space requires 512 bytes. If the configuration SEEPROM device connected to TWI 0 is larger than 512 bytes, the additional space can be accessed as a normal TWI device with the NVMD set to 0000b and the other fields set as documented in Section 7.27, “[GET_NVMD_DATA Command](#)” or Section 7.28, “[SET_NVMD_DATA Command](#)”.

4.3.1 General Configuration Table

[Table 29](#) provides the byte offsets into the configuration SEEPROM where the version number of the SEEPROM binary resides.

Table 28 General Configuration

Byte Offset	Assignment Description	SPC Setting
58	SEEPROM Version	Calculated
59	SEEPROM Version	Calculated
60	SEEPROM Version	Calculated
61	SEEPROM Version	Calculated

4.4 Power-On Self-Test (POST)

From the perspective of the SPC 8x6G bootloader ROM and the ILA firmware, the POST consist of only memory tests, including the CPU instructions and data caches, the CPU’s local memory spaces (PCS_LM), and the SPC 8x6G’s internal memory.

This section specifies all of the memory tests that are used within the bootloader ROM and the ILA. However, not all tests are applicable to all CPUs.

4.4.1 CPU Instruction and Data Caches

The MIPS CPU instruction cache testing consists of writing a known pattern to some memory locations using uncached addresses. The instruction cache is then invalidated entirely. When cache filling is triggered, the last line invalidated is the first line to be refilled. Trigger the cache fill operation in order to copy the content of the data pattern from the RAM into the instruction cache. Finally, read back the instruction cache content via the Index Load Tags instructions and compare against the known pattern.

Data cache testing consists of first mapping the data cache to some physical RAMs using the Index Store Tag Cache instruction. A pattern is then written to the RAM via the data cache normally. This pattern is read back and compared with the original. If the patterns match, the cache test is successful.

4.4.2 Randomly Accessible Memory Tests

Given the large sizes of the memories and the timing constraint on bootloading, the memory testing is not exhaustive. The purpose of the tests is not to ensure that all memory locations are in working order.

This test coverage is actually done during run-time using parity checking. During the POST sequence, the memory tests ensure that the memory spaces are individually accessible and that no two data bus signals are shorted together. To this end, the following tests need to be performed:

- Walking-1 Test – This test ensures that no two data bits within the width of the memory device are tied (high) together. The test successively drives only one data bit high at a time starting with the least significant data bit and ending with the highest order bit of the memory device data bus.
- Walking-0 Test – This test is similar to the Walking-1 test, except that the bit pattern is inverted. That is, one data bit is driven low (0) at a time, while the remaining bits are kept high (1), until the width of the memory device is traversed.
- Ramp Test – This test writes to successive 32-bit word memory locations with an increasing data value. The purpose of this test is to ensure, as best as possible, that no two distinct memory cells are mapped to the same address.

4.5 Flash Memory Format

[Figure 44](#) shows the format of the flash memory for the SPC 8x6G.

Figure 44 Flash Memory Map

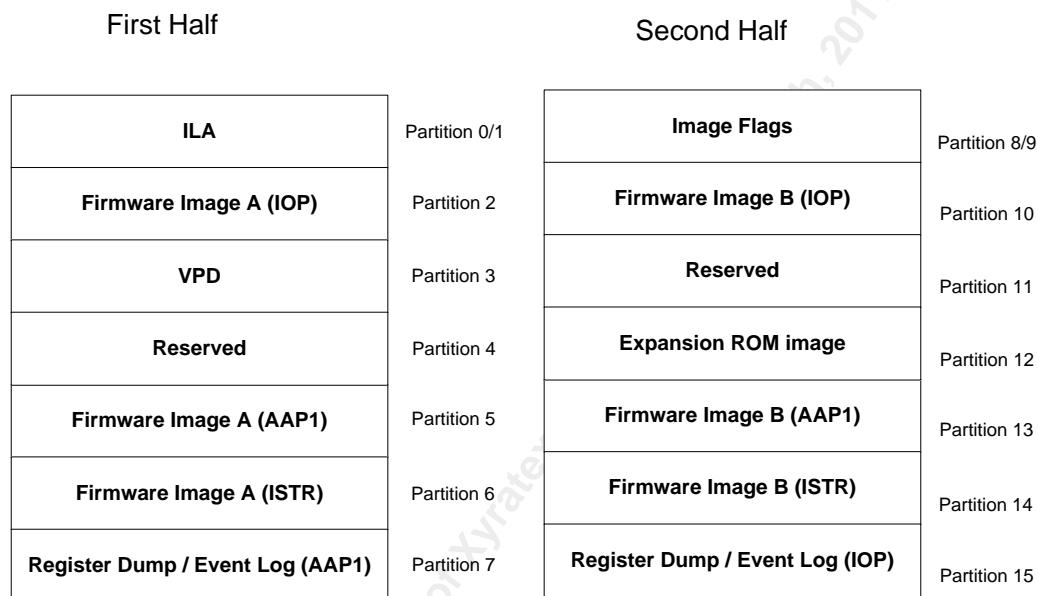


Table 35 Flash Partitions

Partition Number	Partition Name	Partition Start Address	Partition End Address	Current Usage Model	Partition Size
0/1	ILA	0xB800_0000	0xB800_FFFF	64 Kbytes	64 Kbytes
2	IOP - A	0xB801_0000	0xB80A_FFFF	640 Kbytes	640 Kbytes
3	VPD	0xB80B_0000	0xB80F_FFFF	4 Kbytes	320 Kbytes
4	Reserved	0xB810_0000	0xB81F_FFFF	Reserved	1024 Kbytes
5	AAP1 - A	0xB820_0000	0xB827_FFFF	512 Kbytes	512 Kbytes
6	ISTR - A	0xB828_0000	0xB828_FFFF	64 Kbytes	64 Kbytes
7	Reg.Dump AAP1 (at offset 0x0, length 16 Kbytes) Event Log AAP1 (at offset 0x1_0000, length 16 Kbytes)	0xB829_0000	0xB839_FFFF	32 Kbytes	1472 Kbytes
8/9	Image Flags	0xB840_0000	0xB840_FFFF	1 Byte	64 Kbytes
10	IOP- B	0xB841_0000	0xB84A_FFFF	640 Kbytes	640 Kbytes
11	Reserved	0xB84B_0000	0xB84F_FFFF	320 Kbytes	320 Kbytes

Partition Number	Partition Name	Partition Start Address	Partition End Address	Current Usage Model	Partition Size
12	Expansion ROM Image	0xB850_0000	0xB85F_FFFF	Customer defined	1024 Kbytes
13	AAP1- B	0xB860_0000	0xB867_FFFF	512 Kbytes	512 Kbytes
14	ISTR- B	0xB868_0000	0xB868_FFFF	64 Kbytes	64 Kbytes
15	Reg.Dump IOP (at offset 0x0, length 16 Kbytes) Event Log IOP (at offset 0x1_0000, length 16 Kbytes)	0xB869_0000	0xB87F_FFFF	32 Kbytes	1472 Kbytes

The SPC 8x6G supports two firmware partitions, A and B, for each processor and initialization string (ISTR). This enables a mechanism for reliable firmware upgrades.

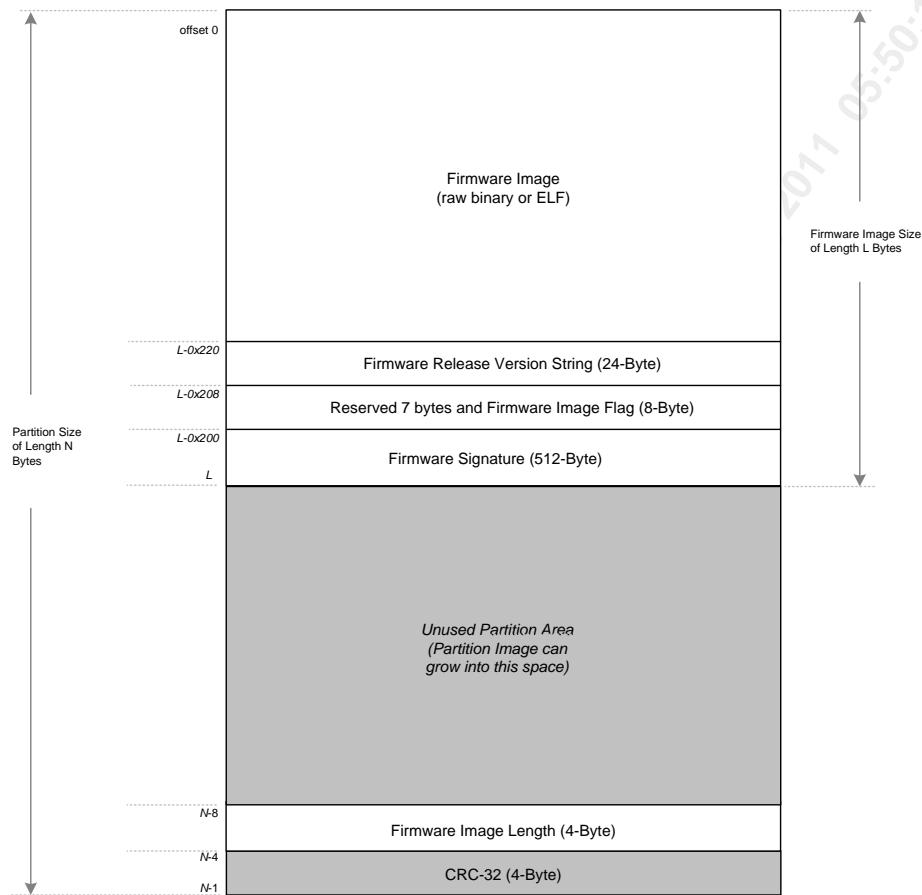
The firmware upgrade mechanism is described in Section [3.15](#).

4.6 Flash Memory Partition Format

The content of the partitions in flash memory is not specified. The partitions can store a firmware image such as an initialization string, an ILA image, an executable application firmware image, or raw binary such as customer-defined VPD, Expansion ROM, a register dump, or an event log, etc.

[Figure 45](#) shows the format of the each partition in flash memory that may store a firmware/partition image.

Figure 45 SPC 8x6G Firmware Image Partition Format



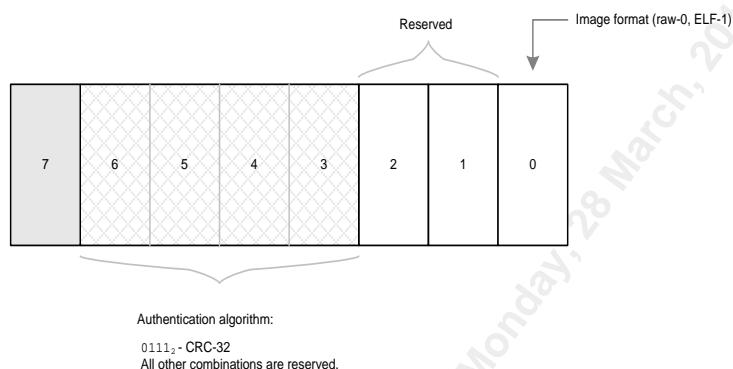
The partition size N must be of a minimum value of 64 KB and larger values at multiples of 64 KB. The top-most 544 bytes of a firmware image are allocated for storing information pertaining to the firmware image. Such information includes the firmware release version string, image flag, and signature for authentication.

The actual byte length of the firmware image is represented by the 4-byte parameter L . The firmware release version string is a 24-byte ASCII string. The location L-0x208 to L-0x202 is the 7-byte reserved field. The location L-0x201 is the firmware image flags byte. The firmware signature sits in the 512-byte authentication storage space. The entire length is located in L-0x200.

The firmware image length is stored with the most significant byte starting at offset $N-8$. The second most significant byte is at offset $N-7$. The second least significant byte is at offset $N-6$. Finally, the least significant byte is stored at offset $N-5$. The last two fields of the partition (the firmware image length and the CRC field) are calculated automatically.

The format of the 1-byte firmware image flags are shown in [Figure 46](#). The most significant bit of this byte is not currently used, and therefore reserved.

Figure 46 Firmware Image Flag Byte Fields



4.7 PCIe System Configuration

After power-up initialization and reading the configuration information from the SEEPROM, the SPC 8x6G boot ROM code initializes the PCIe configuration space. This includes:

- PCI Base Address Register (BAR) configuration
- PCI capabilities configuration
- PCI MSI/MSI-X configuration

Once the PCIe configuration is successfully completed, the SPC 8x6G Boot ROM code releases the SPC 8x6G controller from PCI reset and starts loading the firmware image. Prior to the boot ROM releasing the SPC 8x6G from PCI reset, the SPC 8x6G sends the completion to the configuration request with the status code set to Configuration Request Retry Status. The host should retry the configuration request.

4.8 Firmware Image Load

The SPC 8x6G supports loading firmware images stored in the attached flash memory as well as from the host memory across PCIe interface. The ILA and the firmware image stored in the flash memory are in raw format. The bootloader running on the AAP1 copies the ILA image from flash memory into internal memory and starts using it.

4.9 PCI Configuration Initialization

Once the SPC 8x6G starts responding to PCI configuration transactions, the host can configure the PCI BARs and initialize other PCI Configuration Space registers.

4.10 SPC 8x6G Reset

The following different types of resets are supported for the SPC 8x6G:

- Hard Power-On Reset: This reset is initiated by chip power-on and de-assertion of RST_L. Resets all registers, state machines, firmware and SERDES. The firmware is reloaded. After this reset, the firmware holds the chip in a reset state until the PCIe Configuration Space is initialized. The PCIe Configuration registers and chip registers are not accessible for a 300-millisecond duration. The time required for the firmware to be ready is either 700 milliseconds (UART disabled) or 800 milliseconds (UART enabled).
- Chip Reset: This reset is initiated by the host in response to a fatal error reported by the SPC 8x6G. A chip reset is initiated by writing bit [31], SW_DEVICE_RSTB, of the [SPC Reset Register](#) to zero followed by another write of one to the bit. A chip reset will cause the SPC 8x6G to reinitialize the whole chip, reload the firmware, and cause PCIe reset and PCIe Configuration Space changes. See Section 11.2, “Device Specific Fatal Errors”, Section 11.4, “Device Specific Fatal Error Recovery Procedures”, and Section 11.4.4, “Chip Reset” for more details.
- Soft Reset (normal mode): This reset is initiated by the host in response to a fatal error reported by the SPC 8x6G. A soft reset is initiated by setting the bits in the [SPC Reset Register](#) to selectively reset the SPC 8x6G subsystem components without resetting the PCIe subsystem. A soft reset will cause the SPC 8x6G to partially reinitialize the chip, reload the firmware, and preserve the PCIe Configuration Space. See Section 11.2, “Device Specific Fatal Errors” and Section 11.4, “Device Specific Fatal Error Recovery Procedures” for more details. See Section 11.4.2, “Soft Reset” for the steps to perform a soft reset.
- Soft Reset (HDA Mode): This reset is similar to Soft Reset (normal mode), except that the SPC 8x6G is operating in the Host Direct Access (HDA) mode. See Section 3.21, “Host Direct Access (HDA) Mode” for HDA mode and Section 11.4.3, “Soft Reset Recovery (HDA Mode)” for the steps to perform a soft reset in HDA mode.

4.11 SPC 8x6G Initialization

The following steps are required to initialize the SPC 8x6G:

1. Scan through PCIe Configuration Space to locate the SPC 8x6G and its BAR setting.
2. If operating in Host Direct Access (HDA) mode, complete the HDA mode initialization sequence as described in Section 3.21, “Host Direct Access (HDA) Mode”.

3. Read the MPI Configuration table.
4. Build the data structures for the IQs and OQs.
5. Configure the MPI Configuration table and inform the SPC 8x6G where the data structures are located, as described in Section [5.2](#).

The SPC 8x6G requires one entry in each of the queues to be empty. Therefore, all queues must have a minimum length of two entries. Once the MPI Configuration table is configured, host and the SPC 8x6G firmware can start communicating with each other through the IQs and OQs.

An OQ is assumed to be fully initialized when the host finishes configuring the MPI Configuration table and the SPC 8x6G owns the queue entries and can start sending messages to the host. The host uses the Outbound Queue Consumer Index register to return ownership of processed IOMB in OQ.

The host must allocate one 4-byte word of host memory per IQ and one 4-byte word of host memory per OQ for writing host copies of the current values of Inbound Queue Consumer Index and Outbound Queue Producer Index. The memory address of these locations is written into the MPI Configuration table.

4.11.1 Verifying Host-SPC 8x6G Communication

A host can verify that it can communicate with SPC 8x6G successfully by exchanging [ECHO Command](#) and [ECHO Response](#) IOMBs as described in Sections [7.1](#) and [8.1](#) as follows:

1. Select an inbound circular queue *n* for sending an [ECHO Command](#).
2. Prepare the [ECHO Command](#) in the selected Inbound Circular Queue.
3. Write a new producer index to the SPC 8x6G Inbound Circular Index *n* Producer Index register.

The SPC 8x6G processes the [ECHO Command](#) and sends an [ECHO Response](#) to the outbound circular queue specified by the OBID in the [ECHO Command](#) with the [ECHO Response](#) carrying the same HTAG as specified in [ECHO Command](#) and updates the corresponding outbound circular queue producer index in the host memory.

The host verifies that it receives the [ECHO Response](#) IOMB in the specified outbound circular queue.

4.12 PHY Initialization and Port Instantiation

4.12.1 PHY Initialization

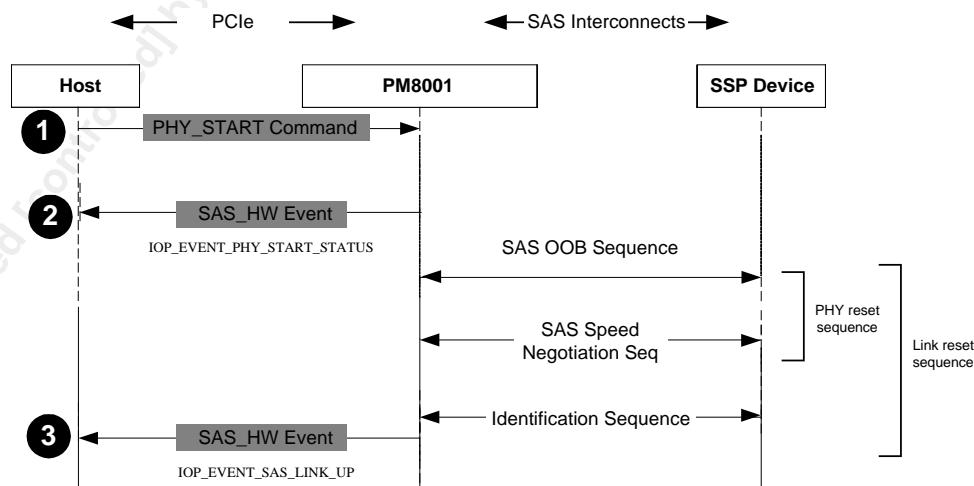
The following sequence describes how the host initializes the individual PHYs on the SPC 8x6G:

1. The host sends a **PHY_START Command** IOMB to the SPC 8x6G. (See Section 7.2, “[PHY_START Command](#)” for details.) In the **PHY_START Command** IOMB, the host must specify the SAS Identify Address Frame and the PHY ID.
2. The SPC 8x6G sends a **SAS_HW_EVENT** Notification IOMB with an event code to indicate the status of the **PHY_START** operation. (See Section 8.2, “[SAS_HW_EVENT Notification](#)” for details.)
3. The SPC 8x6G sends a **SAS_HW_EVENT** Notification IOMB with an indication of one of the following:
 - o SAS Link Up: The remote device is a SAS device and host has received the ID frame of the remote SAS device.
 - o SATA Link Up: The remote device is a SATA device and host has received the initial FIS.
 - o ID Frame Timeout: The SPC 8x6G PHY has failed to complete SAS protocol negotiation with the remote device PHY.

If there is no device/PHY connected to the SPC 8x6G PHY, step 3 does not occur. That is, the SPC 8x6G does not send a notification to the host.

[Figure 47](#) shows the PHY initialization on the SPC 8x6G.

Figure 47 PHY Initialization Flow



During step (1), the host passes the SAS ID Frame in [PHY_START Command](#) IOMB.

During step (2), the host receives the PHY_START acknowledgement and status.

During step (3), the host receives one of the three events: SAS_EVENT_SAS_LINK_UP, SAS_EVENT_SATA_LINK_UP, or SAS_EVENT_ID_FRAME_TIMEOUT. Note that step (3) does not occur if no device is connected to the SPC 8x6G PHY.

4.12.2 Port Instantiation

See Section 3.1, “[SAS Port Instantiation, Port Context and PORT_ID](#)” for the description of how the port is instantiated on the SPC 8x6G.

4.13 Unloading the Host Driver

The host must make sure that communication between it and the SPC 8x6G is terminated before unloading the host driver. To terminate communication, the host must:

1. Stop sending new I/Os to the SPC 8x6G.
2. Wait for all I/Os to complete to make sure that the remote device is in a known state. If the I/Os are not completed in a timely manner, the host should execute the proper cleanup by issuing a task management operation or a LINK/HARD reset PHY control and a local abort (via a [SSP_ABORT Command](#) (Section 7.10) or [SATA_ABORT Command](#) (7.17)).
3. Deregister all device handles using the [DEREGISTER_DEVICE_HANDLE Command](#) described in Section 7.11.
4. Stop the PHYs using the [PHY_STOP Command](#) described in Section 7.3.
5. Terminate MPI communication by issuing the corresponding sequence described in Section 5.2.6.2, “[Host-SPC 8x6G MPI Communication Termination](#)”.

Notes

- Steps (3) to (5) can also be done by issuing a Soft Reset sequence as described in Section 11.4.2, “[Soft Reset Recovery \(Normal Mode\)](#)” or Section 11.4.3, “[Soft Reset Recovery \(HDA Mode\)](#)”.
- The host can skip step (2) above if it is certain that the removed device does not need to be brought into a clean state.

5 Message Passing Interface (MPI) Configuration

Configuration of the Message Passing Interface (MPI) involves several steps to get the host and the SPC 8x6G controller to establish the operating parameters.

The host initiated MPI configuration includes:

- IQ parameters
- OQ parameters
- Interrupt coalescing policy
- Host interrupt property, including PCI legacy interrupt and PCIe MSI-X mapping on the host
- PHY parameters
- Event logging/tracing options
- General status information

5.1 MPI Configuration Table Access

The MPI configuration information is stored in a configuration table, a memory region mapped to the PCIe memory space and accessible by both the host and the SPC 8x6G controller.

In order to access the configuration table, the host needs to know the name and offset of the PCI BAR where the table is accessible. The host gets this information by reading the [Scratchpad 0 Register](#):

- The upper 6 bits [31:26] of the [Scratchpad 0 Register](#) describe the offset within the PCI Configuration Space where the BAR is located.
- The lower 26 bits [25:0] of the [Scratchpad 0 Register](#) describe the offset within the BAR.

Prior to accessing (reading and initializing) the configuration table, the host needs to make sure that the SPC 8x6G firmware is in the ready state. This state is shown in the bits [1:0] of the [Scratchpad 1 Register](#) and the [Scratchpad 2 Register](#), which are set by the AAP1 and the IOP. See [Table 36](#) and [Table 37](#). (For more details about the Scratchpad registers, see Section 10.2, “[Messaging Unit Registers](#)”).

Table 36 Scratchpad 1 Register Status Indication for AAP1

Bits	Fields	Description
[31:8]	AAP_ERR	Fatal error indicators detected on the AAP1/MSGU. Valid only if bits [1:0] of this register are 10b (Error state).
[7:4]	Reserved	Reserved
3	CPU_SOFT_RESET_RDY	AAP1 ready indicator for soft reset.

Bits	Fields	Description
2	SFTRST_P_F	Toggled flag indicating Soft Reset progress.
[1:0]	AAP_STATE	Ready state for the AAP1/MSGU. 00: Power on reset state 01: Soft reset state 10: Error state 11: Ready state

Table 37 Scratchpad 2 Register Status Indication for IOP

Bits	Fields	Description
[31:8]	IOP_ERR	Fatal error indicators detected on the IOP. Valid only if bits [1:0] of this register are 10b (Error state).
[7:4]	Reserved	Reserved
3	CPU_SOFT_RESET_RDY	IOP ready indicator for soft reset.
2	HOST_SOFT_RESET_RDY	Ready indicator for host issuing a soft reset.
[1:0]	IOP_STATE	Ready state for the Input/Output Processing module. 00: Power on reset state 01: Soft reset state 10: Error state 11: Ready state

5.2 MPI Configuration Table

[Table 38](#) lists the main fields defined for the SPC 8x6G MPI configuration table. Each entry is a 32-bit DWord value. The main configuration fields defined in [Table 38](#) also contains offset addresses for general status information as well as additional configuration entries defined for the IQs and OQs.

[Table 39](#) lists the general status for the MPI configuration table as well as the general status of the SPC 8x6G.

[Table 40](#) lists the additional configuration fields for the IQs and [Table 41](#) lists the additional configuration fields for the OQs.

The configuration tables are located in SPC 8x6G memory space.

5.2.1 MPI Main Configuration Table Fields

Table 38 MPI Configuration Table – Main Part

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
0x00	ASCII Signature	AS	SPC 8x6G	Indicate coherent table.	Byte 0: 0x50 (hex value for ASCII character 'P') Byte 1: 0x4D (hex value for ASCII character 'M') Byte 2: 0x43 (hex value for ASCII character 'C') Byte 3: 0x53 (hex value for ASCII character 'S')
0x01	Interface Revision	IR	SPC 8x6G	SPC 8x6G MPI specification number.	
0x02 [3:0]	Firmware Release Type	FWT	SPC 8x6G	Firmware release type: 0x00: Released firmware. 0x01: Development release. 0x02: Alpha release. 0x03: Beta release.	
0x02 [7:4]	Firmware Release Variant	FWV	SPC 8x6G	To be used to identify the compiled variant of a specific firmware version.	
0x02 [15:8]	Firmware Sub-Minor Release Number	FWSMR N	SPC 8x6G	Firmware sub-minor release number. The revision numbering is in decimal format.	
0x02 [23:16]	Firmware Minor Release Number	FWMNR N	SPC 8x6G	Firmware minor release number. The revision numbering is in decimal format.	
0x02 [31:24]	Firmware Major Release Number	FWMJR N	SPC 8x6G	Firmware major release number. The revision numbering is in decimal format.	
0x03	Maximum Outstanding I/O	MOIO	SPC 8x6G	Maximum number of outstanding I/Os supported.	

0x04 [15:0]	Maximum Scatter-Gather List Elements	MSGLE	SPC 8x6G	If non-zero, this is an indication of the maximum number of Scatter/Gather elements (chained or otherwise) that the controller supports in a single request.	
0x04 [31:16]	Maximum Devices	MD	SPC 8x6G	Maximum number of devices connected to the SPC 8x6G. This is the maximum number of device handles for both the target and initiator modes.	
0x05 [7:0]	Maximum Number of IQs	MNIQ	SPC 8x6G	Maximum number of IQs supported. A value of one indicates that the multiple IQs feature is not supported and only a single primary IQ is used.	
0x05 [15:8]	Maximum Number of OQs	MNOQ	SPC 8x6G	Maximum number of OQs supported. A value of one indicates that the multiple OQs feature is not supported and only a single primary OQ is used.	
0x05 [16]	High Priority IQ Support	HPIQS	SPC 8x6G	High priority queue property is supported. 0b: High priority IQs are not supported. 1b: High priority IQs are supported.	
0x05 [17]	Reserved.				
0x05 [18]	Interrupt Coalescing Support	ICS	SPC 8x6G	Support of Interrupt Coalescing mechanisms through firmware. 0b: Interrupt Coalescing mechanisms are not supported. 1b: Interrupt Coalescing mechanisms are supported.	
0x05 [24:19]	Number of PHYs	PN	SPC 8x6G	Number of PHYs.	
0x5 [31:25]	SAS Revision Specification	SASREV	SPC 8x6G	Sets the supported SAS specification: Bit 25: Set if support SAS 1.0. Bit 26: Set if support SAS 1.1. Bit 27: Set if support SAS 2.0. Bits 27-31: Reserved for future use.	

0x06	General Status Table Offset	GSTO	SPC 8x6G	<p>General Status Table (GST) Offset.</p> <p>This is the offset from the first byte of this configuration table to the first byte of the GST defined below.</p>	
0x07	Inbound Queue Configuration Table Offset	IQCTO	SPC 8x6G	<p>IQ Configuration Table (IQCT) Offset.</p> <p>This is the offset from the first byte of this configuration table to the first byte of the IQCT defined below.</p>	
0x08	Outbound Queue Configuration Table Offset	OQCTO	SPC 8x6G	<p>OQ Configuration Table (OQCT) Offset.</p> <p>This is the offset from the first byte of this configuration table to the first byte of the OQCT defined below.</p>	
0x09 [7:0]	Inbound Queue Normal Priority Processing Depth	IQNPPD	Host	<p>When multiple IQs with normal priority are used, this field specifies how many IOMB entries in each normal priority IQ will be fetched before going to the next IQ.</p> <p>0x00: Fetch and process all entries in each normal priority IQ until all entries are fetched, before going to the next normal priority IQ.</p> <p>0x01: Fetch and process one IOMB entry in each normal priority IQ and then fetch another IOMB entry from the next normal priority IQ and so on. This is the strict round-robin processing order.</p> <p>Any other 0xXX value indicates to fetch and process up to 0xXX (the maximum is 0xFF) IOMB entries in each normal priority IQ and then fetch up to 0xXX IOMB entries from the next normal priority IQ and so on.</p>	

0x09 [15:8]	Inbound Queue High Priority Processing Depth	IQHPPD	Host	<p>When multiple IQs with high priority are used, this field specifies how many IOMB entries in each high priority IQ will be fetched before going to the next high priority IQ.</p> <p>0x00: Fetch and process all entries in each high priority IQ until all entries are fetched, before going to the next high priority IQ.</p> <p>0x01: Fetch and process one IOMB entry in each high priority IQ and then fetch another IOMB entry from the next high priority IQ and so on. This is the strict round-robin processing order.</p> <p>Any other 0XX value indicates to fetch and process up to 0XX (the maximum is 0xFF) IOMB entries in each high priority IQ and then fetch up to 0XX IOMB entries from the next high priority IQ and so on.</p>	
0x09 [23:16]	GENERAL EVENT Notification Queue	GENQ	Host	OQ number to receive a GENERAL_EVENT Notification. (See Section 8.18.)	
0x09 [31:24]	DEVICE_HAN DLE_REMOV ED Notification Queue	DHRNQ	Host	OQ number to receive a DEVICE_HANDLE_REMOVE D Notification. (See Section 8.30.)	
0x0A [7:0]	SAS HW Event Notification Queue – PHY_ID 0	SASENQ _P0	Host	OQ number to receive a SAS Hardware event for PHY_ID = 0. (See Section 8.2, “ SAS_HW_EVENT Notification”.)	
0x0A [15:8]	SAS HW Event Notification Queue – PHY_ID 1	SASENQ _P1	Host	OQ number to receive a SAS Hardware event for PHY_ID = 1. (See Section 8.2, “ SAS_HW_EVENT Notification”.)	
0x0A [23:16]	SAS HW Event Notification Queue – PHY_ID 2	SASENQ _P2	Host	OQ number to receive a SAS Hardware event for PHY_ID = 2. (See Section 8.2, “ SAS_HW_EVENT Notification”.)	
0x0A [31:24]	SAS HW Event Notification Queue – PHY_ID 3	SASENQ _P3	Host	OQ number to receive a SAS Hardware event for PHY_ID = 3. (See Section 8.2, “ SAS_HW_EVENT Notification”.)	

0x0B [7:0]	SAS HW Event Notification Queue – PHY _ID 4	SASENQ _P4	Host	OQ number to receive a SAS Hardware event for PHY _ID = 4. (See Section 8.2, “SAS_HW_EVENT Notification“.)	
0x0B [15:8]	SAS HW Event Notification Queue – PHY _ID 5	SASENQ _P5	Host	OQ number to receive a SAS Hardware event for PHY _ID = 5. (See Section 8.2, “SAS_HW_EVENT Notification“.)	
0x0B [23:16]	SAS HW Event Notification Queue – PHY _ID 6	SASENQ _P6	Host	OQ number to receive a SAS Hardware event for PHY _ID = 6. (See Section 8.2, “SAS_HW_EVENT Notification“.)	
0x0B [31:24]	SAS HW Event Notification Queue – PHY _ID 7	SASENQ _P7	Host	OQ number to receive a SAS Hardware event for PHY _ID = 7. (See Section 8.2, “SAS_HW_EVENT Notification“.)	
0x0C [7:0]	SATA NCQ Error Event Notification Queue – PHY _ID 0	SATANE ENQ_P0	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 0. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED_NCQ_MODE (0x00000023).	
0x0C [15:8]	SATA NCQ Error Event Notification Queue – PHY _ID 1	SATANE ENQ_P1	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 1. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED_NCQ_MODE (0x00000023).	
0x0C [23:16]	SATA NCQ Error Event Notification Queue – PHY _ID 2	SATANE ENQ_P2	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 2. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED_NCQ_MODE (0x00000023).	

0x0C [31:24]	SATA NCQ Error Event Notification Queue – PHY _ID 3	SATANE ENQ_P3	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 3. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED _NCQ_MODE (0x00000023).	
0x0D [7:0]	SATA NCQ Error Event Notification Queue – PHY _ID 4	SATANE ENQ_P4	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 4. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED _NCQ_MODE (0x00000023).	
0x0D [15:8]	SATA NCQ Error Event Notification Queue – PHY _ID 5	SATANE ENQ_P5	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 5. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED _NCQ_MODE (0x00000023).	
0x0D [23:16]	SATA NCQ Error Event Notification Queue – PHY _ID 6	SATANE ENQ_P6	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 6. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED _NCQ_MODE (0x00000023).	
0x0D [31:24]	SATA NCQ Error Event Notification Queue – PHY _ID 7	SATANE ENQ_P7	Host	OQ number to receive a SATA error event associated with the NCQ command with PHY _ID = 7. (See Section 8.10, “SATA_EVENT Notification“.) This is to set the default OQ for an NCQ error only, i.e. EVENT is set to IO_XFER_ERROR_ABORTED _NCQ_MODE (0x00000023).	

0x0E [7:0]	I_T Nexus Target Event Notification Queues – PHY _ID 0	ITNTEN Q_P0	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 0. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification “.)	
0x0E [15:8]	I_T Nexus Target Event Notification Queues – PHY _ID 1	ITNTEN Q_P1	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 1. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification “.)	
0x0E [23:16]	I_T Nexus Target Event Notification Queues – PHY _ID 2	ITNTEN Q_P2	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 2. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification “.)	
0x0E [31:24]	I_T Nexus Target Event Notification Queues – PHY _ID 3	ITNTEN Q_P3	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 3. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification “.)	
0x0F [7:0]	I_T Nexus Target Event Notification Queues – PHY _ID 4	ITNTEN Q_P4	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 4. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification “.)	
0x0F [15:8]	I_T Nexus Target Event Notification Queues – PHY _ID 5	ITNTEN Q_P5	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 5. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification “.)	

0x0F [23:16]	I_T Nexus Target Event Notification Queues – PHY _ID 6	ITNTEN Q_P6	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 6. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification”.)	
0x0F [31:24]	I_T Nexus Target Event Notification Queues – PHY _ID 7	ITNTEN Q_P7	Host	In target mode, this field indicates the OQ number to receive I_T Nexus events initiated from a remote initiator device received on PHY_ID = 7. (See Section 8.12, “ DEVICE_HANDLE_ARRIVED Notification”.)	
0x10 [7:0]	SSP Target Event Notification Queue – PHY _ID 0	SSPTEN Q_P0	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY_ID = 0. (See Section 8.13, “ SSP_REQUEST_RECEIVED ”.)	
0x10 [15:8]	SSP Target Event Notification Queue – PHY _ID 1	SSPTEN Q_P1	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY_ID = 1. (See Section 8.13, “ SSP_REQUEST_RECEIVED ”.)	
0x10 [23:16]	SSP Target Event Notification Queue – PHY _ID 2	SSPTEN Q_P2	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY_ID = 2. (See Section 8.13, “ SSP_REQUEST_RECEIVED ”.)	
0x10 [31:24]	SSP Target Event Notification Queue – PHY _ID 3	SSPTEN Q_P3	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY_ID = 3. (See Section 8.13, “ SSP_REQUEST_RECEIVED ”.)	

0x11 [7:0]	SSP Target Event Notification Queue – PHY _ID 4	SSPTEN Q_P4	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY _ID = 4. (See Section 8.13, "SSP_REQUEST_RECEIVED".)	
0x11 [15:8]	SSP Target Event Notification Queue – PHY _ID 5	SSPTEN Q_P5	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY _ID = 5. (See Section 8.13, "SSP_REQUEST_RECEIVED".)	
0x11 [23:16]	SSP Target Event Notification Queue – PHY _ID 6	SSPTEN Q_P6	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY _ID = 6. (See Section 8.13, "SSP_REQUEST_RECEIVED".)	
0x11 [31:24]	SSP Target Event Notification Queue – PHY _ID 7	SSPTEN Q_P7	Host	In target mode, this field indicates the OQ number to receive SSP command requests initiated from a remote initiator device received on PHY _ID = 7. (See Section 8.13, "SSP_REQUEST_RECEIVED".)	
0x12 [15:0]	IO Abort Delay	IOABTDL Y	Host	Maximum delay time in milliseconds before SSP/SATA/SMP COMPLETION response is sent to host from the SPC device. It can be any value except for a zero. A zero value means default: 100 ms. It is used when aborting I/Os with the SSP_ABORT Command, SATA_ABORT Command, SMP_ABORT Command, PORT_CONTROL(PORT_IO_ABORT) Command, and PORT_CONTROL(HARD_RESET) Command.	
0x12 [31:16]	Reserved	—	—	—	

Customization Setting	CUSTSET	Host	Flags for customization. See Section 3.24 for details	
MSGU Event Log Buffer Address Higher	MELBAH	Host	Higher 32 physical address bits of the buffer in host memory where the SPC 8x6G MSGU stores the event log and debug trace information, which is a debug log reported the same as an event log where the severity is set to 0x5 or higher.	
MSGU Event Log Buffer Address Lower	MELBAL	Host	Lower 32 physical address bits of the buffer in host memory where the SPC 8x6G MSGU stores the event log and debug trace information, which is a debug log reported the same as an event log where the severity is set to 0x5 or higher.	
MSGU Event Log Buffer Size	MELBS	Host	Size in bytes of the host memory allocated buffer for SPC 8x6G MSGU event logging. Setting this field to zero indicates that event logging is disabled. If enabled, the minimum size is 1 Kbyte.	0x00

0x17 [3:0]	MSGU Event Log Severity	MELSEV	Host	<p>Bit field option to specify the desired severity level of the SPC 8x6G MSGU event log:</p> <p>0x0: Disable logging.</p> <p>0x1: Critical Error – Any firmware-detected error or top-level error interrupt that disables the SPC 8x6G. Example: Firmware asserts due to an internal condition or interrupt.</p> <p>0x2: Warning – Occurs when a resource is unavailable, an unexpected request occurs, or frames/messages are dropped due to an unavailable handler. Examples:</p> <ul style="list-style-type: none"> • Firmware drops a message from hardware that is unsupported or the handler is not defined. • An incoming target mode request is dropped due to resource unavailability. • The IOP receives a request from the host with an unsupported opcode or Invalid. <p>0x3: Notice – All transport and interconnect events and errors. Examples:</p> <ul style="list-style-type: none"> • Check conditions. • All errors in command completion. • PHY down and PHY up, etc. • TMFs. • All error messages posted from hardware. <p>0x4: Information – Log events in the successful IO Path. Log all IO requests received and completed by the SPC 8x6G. Example: SSP_IO, SATA_IO with host tag, device ID, protocol tag, etc.</p> <p>0x5: Debugging – This is a special level meant to debug consistently reproducible issues by adding more logs while debugging. This level is currently for internal use.</p>
---------------	----------------------------	--------	------	--

0x17 [31:4]	Reserved				
0x18	IOP Event Log Buffer Address Higher	IELBAH	Host	Higher 32 physical address bits of the buffer in host memory where the SPC 8x6G IOP stores the event log and debug trace information, which is a debug log reported the same as an event log where the severity is set to 0x5 or higher.	
0x19	IOP Event Log Buffer Address Lower	IELBAL	Host	Lower 32 physical address bits of the buffer in host memory where the SPC 8x6G IOP stores the event log and debug trace information, which is a debug log reported the same as an event log where the severity is set to 0x5 or higher.	
0x1A	IOP Event Log Buffer Size	IELBS	Host	Size in bytes of the host memory-allocated buffer for SPC 8x6G IOP event logging. Setting this field to zero indicates that event logging is disabled. If enabled, the minimum size is 1 Kbyte.	0x00

0x1B [3:0]	IOP Event Log Severity	IELSEV	Host	<p>Bit field option to specify the desired severity level of the SPC 8x6G IOP event log:</p> <p>0x0: Disable logging.</p> <p>0x1: Critical Error – Any firmware-detected error or top-level error interrupt that disables the SPC 8x6G. Example: Firmware asserts due to an internal condition or interrupt.</p> <p>0x2: Warning – Occurs when a resource is unavailable, an unexpected request occurs, or frames/messages are dropped due to an unavailable handler. Examples:</p> <ul style="list-style-type: none"> • Firmware drops a message from hardware that is unsupported or the handler is not defined. • An incoming target mode request is dropped due to resource unavailability. • The IOP receives a request from the host with an unsupported opcode or Invalid. <p>0x3: Notice – All transport and interconnect events and errors. Examples:</p> <ul style="list-style-type: none"> • Check conditions. • All errors in command completion. • PHY down and PHY up, etc. • TMFs. • All error messages posted from hardware. <p>0x4: Information – Log events in the successful IO Path. Log all IO requests received and completed by the SPC 8x6G. Example: SSP_IO, SATA_IO with host tag, device ID, protocol tag, etc.</p> <p>0x5: Debugging – This is a special level meant to debug consistently reproducible issues by adding more logs while debugging. This level is currently for internal use.</p>	
------------	------------------------	--------	------	---	--

0x1B [23:4]	Reserved				
0x1B [31:24]	Reserved (internal use)			Reserved for internal use.	
0x1C [0]	Fatal Error Interrupt Enable	FERRIE	Host	Flag to indicate whether to enable interrupt when fatal error is detected by SPC 8x6G. When enabled, the field FERRIV specifies which interrupt vector to use. 0b: disable interrupt notification. 1b: enable interrupt notification.	
0x1C [1]	64-bit Addressed PI/CI Support	PCAD64	Host	Flag to indicate whether effective PI/CI address width is 64-bit or not. 0b: effective PI/CI address width is 39 bits. 1b: effective PI/CI address width is 64 bits.	0
0x1C [2]	SGPIO IOMB support	SGPIOIO S	Host	Used by firmware to overwrite SGPIO structures which are initialized with an init string during firmware initialization. If this bit is set to 1, firmware gives full control to the host to send SGPIO IOMB to read/write to SGPIO registers. This bit needs to be set during the initialization stage. If this bit is set to 1, the host still needs to set the GPIO Enable bit by writing to the GPIO_CFG[0] register defined in the SFF-8485 specification before sending other SGPIO IOMBs. If this bit is set to 0, firmware controls the SGPIO registers by setting bits such that the SGPIO LEDs show PHYRDY and I/O activity. This is the default SPC setting. 0x0: FW sets SGPIO registers to indicate PHYRDY and Activity. 0x1: Host controls SGPIO through SGPIO IOMB.	0
0x1C [6:3]	Reserved	Reserved		Reserved	

0x1C [7]	Forced Normal Prioriy on All OQs.	FNPOQ	Host	<p>Outbound queue processing option:</p> <p>0b: This is the default functionality used by all firmware releases. The MSGU firmware will route all high-priority IOMB responses to a high-priority BDMA engine and route all normal-priority IOMB responses to a normal-priority BDMA engine. If high-priority and normal-priority inbound IOMB commands are routed to the same outbound queue, the response IOMBs can be received in that OQ in an order that may not match the order in which the SPC 8x6G processed the commands.</p> <p>1b: The MSGU firmware will route all outbound IOMB responses to a normal-priority BDMA engine. This allows both high-priority and normal-priority inbound IOMBs to be processed and the responses sent to the same OQ so that the host driver knows the order in which the SPC 8x6G processed the commands. Sets all OQ processing to use normal priority to guarantee ordering of the IOMB response.</p>	
0x1C [15:8]	Fatal Error Interrupt Vector	FERRIV	Host	<p>When multiple MSI or MSI-X interrupt vectors are available and enabled, this parameter configures which interrupt vector is used to notify host of the fatal error. See Section 11.2, “Device Specific Fatal Errors” for details. This is a zero-based relative interrupt vector assigned to this instance of the SPC 8x6G.</p>	

0x1C [16]	Enable 64 IQs and 64 OQs	E64Q	Host	<p>This bit enables or disables the support of the upper IQs (IQ32 to IQ63) and upper OQs (OQ32 to OQ63):</p> <p>0: Disable the support of the upper IQs and OQs (support IQ0 to IQ31 and OQ0 to OQ31).</p> <p>1: Enable the support of the upper IQs and OQs (support IQ0 to IQ63 and OQ0 to OQ63). The maximum number of high priority IQs is still limited to 32 and can be assigned anywhere between IQ0 and IQ63.</p> <p>When the support of upper IQs and OQs is enabled, there will be an additional step (step number 8) needed during the MPI Configuration Initialization sequence as described in Section 5.2.6.1, “Host-SPC 8x6G MPI Initialization”.</p>	0
0x1C [17]	Interrupt Reassertion Enable	IRAE	Host	<p>An option that the host can set to cause the SPC 8x6G to re-assert the host interrupt (after the host clears the interrupt) when the PI/CI of an OQ indicates that some outbound IOMB entries have not been consumed by the host after a certain time delay.</p> <p>Interrupt reassertion enable flag:</p> <p>0: Disable</p> <p>1: Enable</p>	
0x1C [18]	Interrupt Reassertion Unit	IRAU	Host	<p>Interrupt reassertion delay unit:</p> <p>0: 10 microsecond increment</p> <p>1: Millisecond increment</p>	

0x1C [31:19]	Interrupt Reassertion Delay	IRAD	Host	The frequency of the interrupt reassertion in the time units specified in the IRAU field above after the SPC 8x6G detects that the host does not completely drain the outbound queue. The reassertion interrupt time is global across all interrupt vectors. It is asserted on a vector and does not occur per outbound queue. The driver is responsible for managing multiple outbound queues per vector. In the case of INTx, this minimizes the number of interrupts received by the host. A value of 0 indicates that the SPC 8x6G is using the default value of 1000 microseconds.	
0x1D	Fatal Error Register Dump Offset for MSGU	FERDO MSGU	SPC 8x6G	The offset of GSM location where the MSGU register dump is located. For details about register dumps, see Section 11.2.15, “Firmware Fatal Errors Register Dump”.	
0x1E	Fatal Error Register Dump Length for MSGU	FERDLM SGU	SPC 8x6G	The length in bytes of the MSGU register dump. For details about register dumps, see Section 11.2.15, “Firmware Fatal Errors Register Dump”.	
0x1F	Fatal Error Register Dump Offset for IOP	FERDOI OP	SPC 8x6G	The offset of GSM location where the IOP register dump is located. For details about register dumps, see Section 11.2.15, “Firmware Fatal Errors Register Dump”.	
0x20	Fatal Error Register Dump Length for IOP	FERDLI OP	SPC 8x6G	The length in bytes of the IOP register dump. For details about register dumps, see Section 11.2.15, “Firmware Fatal Errors Register Dump”.	
0x21 [1:0]	HDA Bootstrap Pins LBI_A[14:13]	HDABSP	SPC 8x6G	The current setting of the bootstrap pins, LBI_A[14:13], used for the HDA boot sequence by the boot ROM. For details of HDA operation, see Section 3.21, “Host Direct Access (HDA) Mode”.	
0x21 [2]	HDA SEEPPROM ‘Force HDA Mode’ Bit	HDASEE PROM	SPC 8x6G	The current setting of the SEEPPROM Configuration byte 4, bit 5, “Force HDA Mode” bit. For details of HDA operation, see Section 3.21, “Host Direct Access (HDA) Mode”.	

0x21 [3]	HDA Firmware	HDAFW	SPC 8x6G	The load method of the SPC 8x6G of firmware: 0: Firmware load from flash memory. 1: Firmware load through HDA mode. For details of HDA operation, see Section 3.21, “ Host Direct Access (HDA) Mode ”.	
0x21 [31:4]	RESERVED				
0x22	SAS PHY Analog Setup Table Offset	SPASTO	SPC 8x6G	Offset of the SAS PHY Analog Setup table defined in Section 5.2.5, “ MPI SAS PHY Analog Setup Table Fields ”. This is the offset from the first byte of this configuration table to the first byte of the SAS PHY Analog Setup table (SPAST) defined below.	

5.2.2 MPI General Status Table Fields

[Table 39](#) lists the SPC 8x6G’s general status information that is included as part of the Configuration Table. The status information provided is an extension of direct PCIe register access as well as information accessed using the IOMB. All fields in the GST are host read-only access.

Table 39 General Status Table (GST)

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
w [2:0]	MPI State	MPI-S	SPC 8x6G	<p>Flag indicating the state of Configuration Table:</p> <p>000: The host has not initialized MPI.</p> <p>001: The host has successfully initialized the MPI.</p> <p>010: Configuration termination in progress. Once it is done, the state will become 000.</p> <p>011: An error has occurred while the host is initializing the MPI Configuration Table. The HMI_ERR field in DWord w bits [31:16] indicates the error status.</p> <p>Other encoding: Reserved.</p>	
[3]	IQ Frozen	QF	SPC 8x6G	The host has frozen one or more IQs. The IFRZ field in DWord w+1 indicates which IQ is in a frozen state.	
w [15:4]	GST Table Length	GSTLEN	SPC 8x6G	Number of DWords in the GST entry including DWord 0.	
w [31:16]	Host MPI Initialization Error	HMI-ERR	SPC 8x6G	<p>This field indicates the details of the error when MPI-S is set to 011:</p> <p>0x0000: Configuration Table OK.</p> <p>[15:10]: Queue Number. This field defines the queue number for errors applicable to the IQ and OQ fields of the MPI Configuration table. Up to 64 IQs/OQs are supported.</p> <p>[9:7]: Table ID. This field defines the four types of table fields supported:</p> <ul style="list-style-type: none"> 000b: Main Configuration Table 001b: GST 010b: IQ Table 011b: OQ Table <p>[6:0]: Error Details. Up to 128 errors are supported for each type of table fields:</p> <p><i>Main Configuration Errors:</i></p> <ul style="list-style-type: none"> 0000001b: SAS event OQ value in the Configuration Table is wrong (queue number is equal to or greater than the number of OQs available). 0000010b: SATA event OQ value in the Configuration Table is wrong (queue number is equal to or greater 	

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
				<p>than the number of OQs available).</p> <p>0000011b: IT event OQ value in the Configuration Table is wrong (queue number is equal to or greater than the number of OQs available).</p> <p>0000100b: SSP event OQ value in the Configuration Table is wrong (queue number is equal to or greater than the number of OQs available).</p> <p>0000110b: AAP1 event log buffer address is wrong (if the buffer size is non-zero, but the addresses are zero).</p> <p>0000111b: IOP event log buffer address is wrong (if the buffer size is non-zero, but the addresses are zero).</p> <p>0001000b: General event OQ value in the Configuration Table is wrong (queue number is equal to or greater than the number of OQs available).</p> <p>0001001b: MPI configuration table re-configuration error. This error is set when the host attempts to initialize the MPI when it is already initialized. If the host wishes to re-initialize the MPI, it should terminate the MPI communication by following the MPI communication termination procedure (section 5.2.6.2). If this error is observed it is recommended that the host issue a softreset prior to attempting to initialize the MPI.</p> <p><i>GST Table Errors:</i></p> <p>N/A</p> <p><i>IQ Table Errors:</i></p> <p>0000001b: IQ wrong priority (priority is not 00 or 01).</p> <p>0000010b: IQ wrong element size (element size is 0 or not 32-byte aligned).</p> <p>0000011b: IQ wrong CI host address (address is 0).</p> <p>0000100b: IQ wrong base host address (address is 0).</p> <p>0000101b: No IQ enabled (at least one IQ must be enabled).</p> <p>0000110b: Exceeded the maximum of 32 High Priority IQs.</p>	

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
				OQ Table Errors: 0000001b: OQ wrong element size (element size is 0 or not 32-byte aligned). 0000010b: OQ wrong PI host address (address is 0). 0000011b: OQ wrong queue base host address (address is 0). 0000100b: No OQ enabled (at least one OQ must be enabled).	
w+1	Inbound Queue Freeze State 0	IFRZ0	SPC 8x6G	Freeze status for the first set of 32 IQs. Each bit represents the IQ freeze state. Bit 0 represents IQ 0 and bit 31 represents IQ 31.	
w+2	Inbound Queue Freeze State 1	IFRZ1	SPC 8x6G	Freeze status for the second set of 32 IQs. Each bit represents the IQ freeze state. Bit 0 represents IQ 32 and bit 31 represents IQ 63.	
w+3	SPC 8x6G MSGU Tick Count	MSGUTCNT	SPC 8x6G	Ticks (in seconds) since the MSGU comes out of the reset state. This field is incremented by the SPC 8x6G every second. It is reset back to zero when it overflows.	
w+4	SPC 8x6G IOP Tick Count	IOPTCNT	SPC 8x6G	Ticks (in seconds) since the IOP comes out of reset. This field is incremented by the SPC 8x6G every second. It is reset back to zero when it overflows.	
w+5	Reserved				
w+6 [0]	PHY Start State 0	PHYSTAR T0	SPC 8x6G	PHY-0 start state: 0: PHY not started 1: PHY started	
w+6 [1]	PHY Link State 0	PHYLINK 0	SPC 8x6G	Valid only if PHY-0 is started. PHY-0 link state: 0: PHY link is down 1: PHY link is up	

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
w+6 [31:2]	Reserved				
w+7 [0]	PHY Start State 1	PHYSTAR T1	SPC 8x6G	PHY-1 start state: 0: PHY not started 1: PHY started	
w+7 [1]	PHY Link State 1	PHYLINK 1	SPC 8x6G	Valid only if PHY-1 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+7 [31:2]	Reserved				
w+8 [0]	PHY Start State 2	PHYSTAR T2	SPC 8x6G	PHY-2 start state: 0: PHY not started 1: PHY started	
w+8 [1]	PHY Link State 2	PHYLINK 2	SPC 8x6G	Valid only if PHY-2 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+8 [31:2]	Reserved				
w+9 [0]	PHY Start State 3	PHYSTAR T3	SPC 8x6G	PHY-3 start state: 0: PHY not started 1: PHY started	
w+9 [1]	PHY Link State 3	PHYLINK 3	SPC 8x6G	Valid only if PHY-3 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+9 [31:2]	Reserved				
w+0xA [0]	PHY Start State 4	PHYSTAR T4	SPC 8x6G	PHY-4 start state: 0: PHY not started 1: PHY started	
w+0xA [1]	PHY Link State 4	PHYLINK 4	SPC 8x6G	Valid only if PHY-4 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+0xA [31:2]	Reserved				
w+0xB [0]	PHY Start State 5	PHYSTAR T5	SPC 8x6G	PHY-5 start state: 0: PHY not started 1: PHY started	

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
w+0x0B [1]	PHY Link State 5	PHYLINK 5	SPC 8x6G	Valid only if PHY-5 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+0x0B [31:2]	Reserved				
w+0x0C [0]	PHY Start State 6	PHYSTAR T6	SPC 8x6G	PHY-6 start state: 0: PHY not started 1: PHY started	
w+0x0C [1]	PHY Link State 6	PHYLINK 6	SPC 8x6G	Valid only if PHY-6 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+0x0C [31:2]	Reserved				
w+0x0D [0]	PHY Start State 7	PHYSTAR T7	SPC 8x6G	PHY-7 start state: 0: PHY not started 1: PHY started	
w+0x0D [1]	PHY Link State 7	PHYLINK 7	SPC 8x6G	Valid only if PHY-7 is started. PHY link state: 0: PHY link is down 1: PHY link is up	
w+0x0D [31:2]	Reserved		SPC 8x6G		
w+0x0E	GPIO Input Value	GPIOIV	SPC 8x6G	Status of the GPIO input pins [19:0]. The SPC 8x6G firmware updates the value by once per second after MPI-S is in the ready state (0x1). Bits [31:20] are reserved for future use.	
w+0x0F	Reserved		SPC 8x6G		
w+0x10	Reserved		SPC 8x6G		
w+0x11	Recoverable Error Information 0	RERRINF O0	SPC 8x6G	Recoverable error information 0. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x12	Recoverable Error Information 1	RERRINF O1	SPC 8x6G	Recoverable error information 1. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x13	Recoverable	RERRINF	SPC	Recoverable error information 2.	

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
	Error Information 2	O2	8x6G	For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x14	Recoverable Error Information 3	RERRINF O3	SPC 8x6G	Recoverable error information 3. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x15	Recoverable Error Information 4	RERRINF O4	SPC 8x6G	Recoverable error information 4. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x16	Recoverable Error Information 5	RERRINF O5	SPC 8x6G	Recoverable error information 5. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x17	Recoverable Error Information 6	RERRINF O6	SPC 8x6G	Recoverable error information 6. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	
w+0x18	Recoverable Error Information 7	RERRINF O7	SPC 8x6G	Recoverable error information 7. For details about the error information, see Section 11.3, “Device Specific Recoverable/Correctable Errors”.	

5.2.3 MPI Inbound Queue Configuration Table Fields

[Table 40](#) below lists the configuration parameters for the IQs. The maximum number of supported IQs is described by the MNIQ field offset of the main configuration fields in [Table 38](#). The host must not configure and write beyond the valid/supported IQs.

Table 40 Inbound Queue Configuration Table (IQCT)

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
x [15:0]	Inbound Queue 0 Depth	IQD0	Host	Maximum number of elements in IQ 0. A value of zero indicates that the queue is disabled. Typically, IQ 0 is used as the default IQ and should be enabled.	N/A. Host needs to set.
x [29:16]	Inbound Queue 0 Element Size	IQES0	Host	Size of each queue element in bytes for IQ 0. The size of the queue entry must be a multiple of 32 bytes to match the SPC 8x6G internal cache-line. The host should set the size according to the anticipated IOMB and IOMB size that will be sent through this IQ.	N/A. Host needs to set.
x [31:30]	Inbound Queue 0 Priority	IQP0	Host	IQ 0 priority: 00b: Normal priority. 01b: High priority. 10b: Reserved. 11b: Reserved. When the 'Enable 64 IQs and 64 OQs' (E64Q) in Table 38 "MPI Main Configuration Table Fields" , DWord 0x1C [16] is set to 1, it indicates that support for up to 64 IQs is enabled. The maximum number of high priority IQs is still limited to 32 and can be assigned anywhere between IQ0 and IQ63.	N/A. Host needs to set.
x+1	Inbound Queue 0 Base Address High	IQBAH0	Host	Upper 32 address bits of the host memory region containing the queue elements for IQ 0.	N/A. Host needs to set.
x+2	Inbound Queue 0 Base Address Low	IQBAL0	Host	Lower 32 address bits of the host memory region containing the queue elements for IQ 0.	N/A. Host needs to set.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
x+3	Inbound Queue 0 Consumer Index Base Address High	IQCIBAH0	Host	Upper 32 address bits of the host memory region containing the Consumer Index for IQ 0. The host is responsible for setting the corresponding IQ Consumer Index to zero prior to MPI Initialization as the SPC 8x6G will not initialize it. The SPC 8x6G will use the full address to update the Consumer Index.	N/A. Host needs to set.
x+4	Inbound Queue 0 Consumer Index Base Address Low	IQCIBAL0	Host	Lower 32 address bits of the host memory region containing the Consumer Index for IQ 0. The host is responsible for setting the corresponding IQ Consumer Index to zero prior to MPI Initialization as the SPC 8x6G will not initialize it. The SPC 8x6G will use the full address to update the Consumer Index.	N/A. Host needs to set.
x+5	Inbound Queue 0 Producer Index PCI BAR	IQPIBAR0	SPC 8x6G	The SPC 8x6G PCI BAR location where the IQ 0 PI is located. The SPC 8x6G BAR location is described as the offset from the start of the PCI Configuration space.	No default. Dynamic.
x+6	Inbound Queue 0 Producer Index PCI BAR Offset	IQPIOFF0	SPC 8x6G	The offset within the PCI BAR where the IQ 0 PI is located.	No default. Dynamic.
x+7	Reserved				
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
(x+8) + (m-1)*8 [15:0]	Inbound Queue <i>m</i> Depth	IQD <i>m</i>	Host	Maximum number of elements in IQ <i>m</i> .	N/A. Host needs to set.
(x+8) + (m-1)*8 [29:16]	Inbound Queue <i>m</i> Element Size	IQES <i>m</i>	Host	Size of each queue element in bytes for IQ <i>m</i> . The size of the queue entry must be a multiple of 32 bytes to match the SPC 8x6G internal cache-line. The host should set the size according to the anticipated IOMB and IOMB size that will be sent through this IQ.	N/A. Host needs to set.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
(x+8) + (m-1)*8 [31:30]	Inbound Queue <i>m</i> Priority	IQPM	Host	IQ <i>m</i> priority: 00b: Normal priority. 01b: High priority. 10b: Reserved. 11b: Reserved. When the 'Enable 64 IQs and 64 OQs' (E64Q) in Table 38 "MPI Main Configuration Table Fields" , DWord 0x1C [16] is set to 1, it indicates that support for up to 64 IQs is enabled. The maximum number of high priority IQs is still limited to 32 and can be assigned anywhere between IQ0 and IQ63.	N/A. Host needs to set.
(x+8) + (m-1)*8 + 1	Inbound Queue <i>m</i> Base Address High	IQBAPHm	Host	Upper 32 address bits of the host memory region containing the queue elements for IQ <i>m</i> .	N/A. Host needs to set.
(x+8) + (m-1)*8 + 2	Inbound Queue <i>m</i> Base Address Low	IQBALm	Host	Lower 32 address bits of the host memory region containing the queue elements for IQ <i>m</i> .	N/A. Host needs to set.
(x+8) + (m-1)*8 + 3	Inbound Queue <i>m</i> Consumer Index Base Address High	IQCIBAHm	Host	Upper 32 address bits of the host memory region containing the Consumer Index for IQ <i>m</i> . The host is responsible for setting the corresponding IQ consumer index to zero prior to MPI Initialization as the SPC 8x6G will not initialize it.	N/A. Host needs to set.
(x+8) + (m-1)*8 + 4	Inbound Queue <i>m</i> Consumer Index Base Address Low	IQCIBALm	Host	Lower 32 address bits of the host memory region containing the Consumer Index for IQ <i>m</i> . The host is responsible for setting the corresponding IQ consumer index to zero prior to MPI Initialization as the SPC 8x6G will not initialize it.	N/A. Host needs to set.
(x+8) + (m-1)*8 + 5	Inbound Queue <i>m</i> Producer Index PCI BAR	IQPIBARm	SPC 8x6G	The SPC 8x6G PCI BAR location where the IQ <i>m</i> PI is located. The SPC 8x6G BAR location is described as the offset from the start of the PCI Configuration space.	No default. Dynamic.
(x+8) + (m-1)*8 + 6	Inbound Queue <i>m</i> Producer Index PCI BAR Offset	IQPIOFFm	SPC 8x6G	The offset within the PCI BAR where the IQ <i>m</i> PI is located.	No default. Dynamic.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
(x+8) + (m-1)*8 + 7	Reserved				

5.2.4 MPI Outbound Queue Configuration Table Fields

Table 41 below lists the properties of OQs. The maximum number of supported OQs is described by the MNOQ field offset of the main configuration fields in Table 38. The host must not configure and write beyond the valid/supported OQs.

Table 41 Outbound Queue Configuration Table (OQCT)

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
y [15:0]	Outbound Queue 0 Depth	OQD0	Host	Maximum number of elements in OQ 0. A value of zero indicates that the queue is disabled. Typically, OQ 0 is used as the default OQ and should be enabled.	N/A. Host needs to set.
y [29:16]	Outbound Queue 0 Element Size	OQES0	Host	Size of each queue element in bytes for OQ 0. The size of the queue entry needs to be multiple of 32 bytes to match SPC 8x6G internal cache-line. The minimum size is 64 bytes.	N/A. Host needs to set.
y [30]	Outbound Queue 0 Interrupt Enable	OQIE0	Host	Interrupt Enable for OQ 0. 0b: No interrupt to host (host polling) 1b: Interrupt enabled.	N/A. Host needs to set.
y [31]	Reserved				
y+1	Outbound Queue 0 Base Address High	OQBAH0	Host	Upper 32 address bits of the host memory region containing the queue elements (circular buffer) for OQ 0.	N/A. Host needs to set.
y+2	Outbound Queue 0 Base Address Low	OQBAL0	Host	Lower 32 address bits of the host memory region containing the queue elements (circular buffer) for OQ 0.	N/A. Host needs to set.
y+3	Outbound Queue 0 Producer Index Base Address High	OQPIBAH0	Host	Upper 32 address bits of the host memory region containing the PI for the OQ 0.	N/A. Host needs to set.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
y+4	Outbound Queue 0 Producer Index Base Address Low	OQPIBAL0	Host	Lower 32 address bits of the host memory region containing the PI for the OQ 0.	N/A. Host needs to set.
y+5	Outbound Queue 0 Consumer Index PCI BAR	OQCIBAR0	SPC 8x6G	The SPC 8x6G PCI BAR location where the OQ 0 CI is located. This is also the PCI BAR number for OQDICTOFF0. The SPC 8x6G BAR location is described as the offset from the start of the PCI Configuration space.	No default. Dynamic.
y+6	Outbound Queue 0 Consumer Index PCI BAR Offset	OQCIOFF0	SPC 8x6G	The offset within the PCI BAR where the OQ 0 CI is located.	No default. Dynamic.
y+7 [15:0]	Outbound Queue 0 Interrupt Coalescing Timeout	OQICT0	Host	Maximum time, in 10 microseconds, from the instant an unmasked OQ 0 interrupt event occurs to the time the SPC 8x6G asserts the interrupt signal. A zero value means not to delay the interrupt signal assertion. An interrupt signal is asserted when either of the OQICT or OQICC (below) parameters is satisfied.	0x00

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
y+7 [23:16]	Outbound Queue 0 Interrupt Coalescing Count	OQICCO	Host	<p>Maximum number of unmasked OQ0 interrupt events accumulated (coalesced) before the SPC 8x6G asserts the interrupt signal. A value of 0 or 1 indicates no coalescing.</p> <p>If this parameter is non-zero but the OQICTO parameter is set to zero, the SPC 8x6G automatically sets the OQICTO to a 100 microsecond timeout. This is implemented to guarantee that the interrupt will be generated when the OQICCO condition may not ever be satisfied.</p> <p>An interrupt signal is asserted when either of the OQICT (above) or OQICC parameters is satisfied.</p>	0
y+7 [31:24]	Outbound Queue 0 Interrupt Vector	OQIV0	Host	When MSI or MSI-X interrupt modes are enabled this parameters configures which interrupt vector is used for OQ 0. This is a zero-based relative interrupt vector assigned to this instance of SPC 8x6G. If there is only a single interrupt vector (MSI, MSI-X, or legacy INT-x) assigned to this instance of SPC 8x6G, set this value to zero.	N/A. Host needs to set.
y+8	Outbound Queue 0 Dynamic Interrupt Coalescing Timeout	OQDICTOF F0	SPC 8x6G / host	<p>The offset within the PCI BAR where the Dynamic Interrupt Coalescing Timeout variable is located. The PCI BAR number is indicated by OQCIBAR0 field.</p> <p>The initial value is set by the SPC 8x6G based on OQICT0 set by the host. During run-time the host could dynamically change the Interrupt Coalescing Timeout by writing the new value to this location. The unit is in 10 microseconds similar to OQICT0.</p>	Default set based on OQICT0. The host can change dynamically during run-time I/O.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
y+9 + (n-1)*9 [15:0]	Outbound Queue <i>n</i> Depth	OQD <i>n</i>	Host	Maximum number of Elements in OQ <i>n</i> . A value of zero indicates that the queue is disabled.	N/A. Host needs to set.
y+9 + (n-1)*9 [29:16]	Outbound Queue <i>n</i> Element Size	OQES <i>n</i>	Host	Size of each queue element in bytes for OQ <i>n</i> . The size of the queue entry must be a multiple of 32 bytes to match the SPC 8x6G internal cache-line. The minimum size is 64 bytes.	N/A. Host needs to set.
y+9 + (n-1)*9 [30]	Outbound Queue <i>n</i> Interrupt Mode	OQIM <i>n</i>	Host	Interrupt Mode for OQ <i>n</i> . 0b: No interrupt to host (host polling). 1b: Interrupt enabled.	N/A. Host needs to set.
y+9 + (n-1)*9 [31]				Bits [31]: Reserved	0x00
y+9 + (n-1)*9 +1	Outbound Queue <i>n</i> Base Address High	OQBAH <i>n</i>	Host	Upper 32 address bits of the host memory region containing the queue elements (circular buffer) for OQ <i>n</i> .	N/A. Host needs to set.
y+9 + (n-1)*9 +2	Outbound Queue <i>n</i> Base Address Low	OQBAL <i>n</i>	Host	Lower 32 address bits of the host memory region containing the queue elements (circular buffer) for OQ <i>n</i> .	N/A. Host needs to set.
y+9 + (n-1)*9 +3	Outbound Queue <i>n</i> Producer Index Base Address High	OQPBAH <i>n</i>	Host	Upper 32 address bits of the host memory region containing the PI for OQ <i>n</i> .	N/A. Host needs to set.
y+8 + (n-1)*9 +4	Outbound Queue <i>n</i> Producer Index Base Address Low	OQPBAL <i>n</i>	Host	Lower 32 address bits of the host memory region containing the PI for OQ <i>n</i> .	N/A. Host needs to set.
y+9 + (n-1)*9 +5	Outbound Queue <i>n</i> Consumer Index PCI BAR	OQCIBAR <i>n</i>	SPC 8x6G	The SPC PCI BAR number where the OQ <i>n</i> CI is located. This is also the PCI BAR number for OQDICTIONOFF <i>n</i> . The SPC 8x6G BAR location is described as the offset from the start of the PCI Configuration space.	No default. Dynamic.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
y+9 + (n-1)*9 + 6	Outbound Queue <i>n</i> Consumer Index PCI BAR Offset	OQCIOFFn	SPC 8x6G	The SPC 8x6G PCI BAR location where the OQ <i>n</i> CI is located. The SPC 8x6G BAR location is described as the offset from the start of the PCI Configuration space.	No default. Dynamic.
y+9 + (n-1)*9 + 7 [15:0]	Outbound Queue <i>n</i> Interrupt Coalescing Timeout	OQICTn	Host	<p>Maximum time, in 10 microseconds, the interrupt assertion interval from the last interrupt assertion to the host, to the next interrupt assertion, and to the host.</p> <p>A zero value means not to delay the interrupt signal assertion.</p> <p>An interrupt signal is asserted when either of the OQICT or OQICC (below) parameters is satisfied.</p> <p>When multiple OQs are sharing the same interrupt vector, this value should be set to the same value for all these OQs.</p>	0x00
y+9 + (n-1)*8 + 7 [23:16]	Outbound Queue <i>n</i> Interrupt Coalescing Count	OQICCn	Host	<p>Maximum number of unmasked OQ <i>n</i> interrupt events accumulated (coalesced) before the SPC 8x6G asserts the interrupt signal. A value of 0 or 1 indicates no coalescing.</p> <p>An interrupt signal is asserted when either of the OQICT (above) or OQICC parameters is satisfied.</p>	0
y+9+ (n-1)*8 + 7 [31:24]	Outbound Queue <i>n</i> Interrupt Vector	OQIVn	Host	When MSI or MSI-X interrupt modes are enabled, this parameter configures which interrupt vector is used for OQ <i>n</i> . This is a zero-based relative interrupt vector assigned to this instance of the SPC 8x6G. If there is only a single interrupt vector (MSI, MSI-X, or legacy INT-x) assigned to this instance of the SPC 8x6G, set this value to zero.	N/A. Host needs to set.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
y+9+ (n-1)*8 + 8	Outbound Queue <i>n</i> Dynamic Interrupt Coalescing Timeout PCI BAR Offset	OQDICTOF Fn	SPC 8x6G / host	The offset within the PCI BAR where the Dynamic Interrupt Coalescing Timeout variable is located. The PCI BAR number is indicated by OQCIBARn field. The initial value is set by the SPC 8x6G based on the OQICTn set by the host. During run-time, the host could dynamically change the Interrupt Coalescing Timeout by writing the new value to this location. The unit is in 10 microseconds similar to OQICTn.	Default set based on OQICTn. Host can change dynamically during run time I/O.

5.2.5 MPI SAS PHY Analog Setup Table Fields

Table 42 lists the properties of SAS PHY Analog Setup table. It consists of ten groups (or indexes) of table entries. Each group contains ten register values. Currently, there are eight registers specified. This table is used in conjunction with the [PHY_START Command](#) described in Section 7.2.

This table needs to be initialized only if the host intends to use the “ASE” option in the [PHY_START Command](#). The “SPASTI” field in the [PHY_START Command](#) specifies which index in the table is used to configure the PHY analog setting.

Table 42 SAS PHY Analog Setup Table (SPAST)

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
z	Index 0 SAS PHY Analog Register 0	IDX0_SPA_REG0	Host	Value to be set to the OSSP “Transmitter Per Port Configuration 1 SAS_SATA G1 Register”. See Section 10.5.4.	N/A. Host needs to set.
z + 1	Index 0 SAS PHY Analog Register 1	IDX0_SPA_REG1	Host	Value to be set to the OSSP “Transmitter Per Port Configuration 1 SAS_SATA G2 Register”. See Section 10.5.5.	N/A. Host needs to set.
z + 2	Index 0 SAS PHY Analog Register 2	IDX0_SPA_REG2	Host	Value to be set to the OSSP “Transmitter Per Port Configuration 1 SAS_SATA G3 Register”. See Section 10.5.6.	N/A. Host needs to set.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
z + 3	Index 0 SAS PHY Analog Register 3	IDX0_SPA_REG3	Host	Value to be set to the OSSP "Transmitter Configuration 1 Register". See Section 10.5.10.	N/A. Host needs to set.
z + 4	Index 0 SAS PHY Analog Register 4	IDX0_SPA_REG4	Host	Value to be set to the OSSP "Receiver Per Port Configuration 1 SAS_SATA G1G2 Register". See Section 10.5.7.	N/A. Host needs to set.
z + 5	Index 0 SAS PHY Analog Register 5	IDX0_SPA_REG5	Host	Value to be set to the OSSP "Receiver Per Port Configuration 1 SAS_SATA G3 Register". See Section 10.5.8.	N/A. Host needs to set.
z + 6	Index 0 SAS PHY Analog Register 6	IDX0_SPA_REG6	Host	Value to be set to the OSSP "Receiver Configuration 1 Register" See Section 10.5.11.	N/A. Host needs to set.
z + 7	Index 0 SAS PHY Analog Register 7	IDX0_SPA_REG7	Host	Value to be set to the OSSP "Receiver Configuration 2 Register". See Section 10.5.12.	N/A. Host needs to set.
z + 8	Reserved	Reserved	Host	Reserved	N/A.
z + 9	Reserved	Reserved	Host	Reserved	N/A.
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
z +90	Index 9 SAS PHY Analog Register 0	IDX9_SPA_REG0	Host	Value to be set to the OSSP "Transmitter Per Port Configuration 1 SAS_SATA G1 Register". See Section 10.5.4.	N/A. Host needs to set.
z +90+ 1	Index 9 SAS PHY Analog Register 1	IDX9_SPA_REG1	Host	Value to be set to the OSSP "Transmitter Per Port Configuration 1 SAS_SATA G2 Register". See Section 10.5.5.	N/A. Host needs to set.
z +90 +2	Index 9 SAS PHY Analog Register 2	IDX9_SPA_REG2	Host	Value to be set to the OSSP "Transmitter Per Port Configuration 1 SAS_SATA G3 Register". See Section 10.5.6.	N/A. Host needs to set.

DWord [Bit(s)]	Field Name	Field Symbol	Write Access	Description	Default Value
z +90 +3	Index 9 SAS PHY Analog Register 3	IDX9_SPA_REG3	Host	Value to be set to the OSSP "Transmitter Configuration 1 Register". See Section 10.5.10.	N/A. Host needs to set.
z +90+ 4	Index 9 SAS PHY Analog Register 4	IDX9_SPA_REG4	Host	Value to be set to the OSSP "Receiver Per Port Configuration 1 SAS_SATA G1G2 Register". See Section 10.5.7.	N/A. Host needs to set.
z +90+ 5	Index 9 SAS PHY Analog Register 5	IDX9_SPA_REG5	Host	Value to be set to the OSSP "Receiver Per Port Configuration 1 SAS_SATA G3 Register". See Section 10.5.8.	N/A. Host needs to set.
z +90+ 6	Index 9 SAS PHY Analog Register 6	IDX9_SPA_REG6	Host	Value to be set to the OSSP "Receiver Configuration 1 Register" See Section 10.5.11.	N/A. Host needs to set.
z +90+ 7	Index 9 SAS PHY Analog Register 7	IDX9_SPA_REG7	Host	Value to be set to the OSSP "Receiver Configuration 2 Register". See Section 10.5.12.	N/A. Host needs to set.
z +90+ 8	Reserved	Reserved	Host	Reserved	N/A.
z +90+ 9	Reserved	Reserved	Host	Reserved	N/A.

5.2.6 MPI State and Configuration Table Reading and Writing Sequence

The following subsections describe operations related to the host and SPC 8x6G messaging. Operations are initiated by the host by writing and reading a specific bit in the [Inbound Doorbell Register](#).

Only one operation can be done at a time (no OR-ing of bits in the [Inbound Doorbell Register](#)) and only certain operations can be done on a given MSGU state. That is, the host cannot freeze an un-initialized MPI Configuration Table.

5.2.6.1 Host-SPC 8x6G MPI Initialization

This section describes the sequence to establish MPI communication between a host and the SPC 8x6G. At the end of the initialization sequence, the host can send an inbound IOMB command to the SPC 8x6G and the SPC 8x6G can access the negotiated host memory and send an outbound IOMB.

The sequence of reading and writing to the configuration table during initialization is:

1. The host accesses the configuration table using the method described in Section 5.1, “[MPI Configuration Table Access](#)”.
2. To confirm configuration table coherency, the host needs to make sure that the SPC 8x6G firmware is in ready state. Section 5.1 describes the firmware ready state indication through bits [1:0] of the [Scratchpad 1 Register](#) and the [Scratchpad 2 Register](#). The host may need to do some polling for ready state by reading these registers. (See Sections 10.2.6 and 10.2.7 for details about these registers.)
3. Host must ensure that MPI is not already initialized by reading MPI-S field in the [General Status Table \(GST\)](#) described in Section 5.2.2. The expected MPI-S value is 000. MPI re-initialization without prior communication termination results in MPI initialization error described in the HMI_ERR field in the [General Status Table \(GST\)](#).
4. The host reads the configuration table entries with the SPC 8x6G controller write access (offset 0x00 to offset 0x08, offset 0x1D to offset 0x22) to know the controller’s capabilities and limitations.
5. The host writes/initializes the appropriate configuration fields. Please note that the host should only write/initialize the fields that have the host write access defined in [Table 38](#), [Table 40](#), and [Table 41](#).
6. The host notifies the SPC 8x6G controller by setting bit 0 of the [Inbound Doorbell Register](#) as described in Section 10.2.1. The SPC 8x6G MSGU handles the configuration table changes and re-initializes the MSGU and the corresponding hardware blocks.
7. The host reads back the [Inbound Doorbell Register](#) (by polling for up to 2 second) until register bit 0 is cleared indicating that the SPC 8x6G controller has acknowledged the completion of initialization sequence.
8. The host checks for successful MPI initialization by reading the MPI-S field in the [General Status Table \(GST\)](#). An encoding of 001b in the MPI-S field indicates that the host has successfully initialized the MPI.

Note:

- The following step is needed only if support for beyond 32 IQs and 32 OQs is enabled. If not enabled, ignore this step. The support of 64 IQs/64OQs is enabled by setting E64Q flag in DWord 0x1C [16] in [Table 38](#), “[MPI Configuration Table – Main Part](#)”.

9. The host reads back the value of all IQ Producer Index PCI BAR offsets (IQPIOFFn) for all the configured IQs as the SPC 8x6G will remap their addresses. (See [Table 40](#) in Section [5.2.3](#), “[MPI Inbound Queue Configuration Table Fields](#)” for details about IQCT and IQPIOFFn.)

5.2.6.2 Host-SPC 8x6G MPI Communication Termination

This section describes the sequence to terminate MPI communication between a host and the SPC 8x6G that was established at the completion of the sequence described in Section [5.2.6.1](#), “[Host-SPC 8x6G MPI Initialization](#)” above.

MPI communication termination is a step that the host driver is required to execute prior to unloading the driver to notify the SPC 8x6G MSGU to:

- Not access the host memory space following the termination sequence.
- Flush out any unprocessed inbound IOMB.
- Flush out any pending outbound processing.

At the end of the termination sequence, the host will not send an inbound IOMB command to the SPC 8x6G and the SPC 8x6G will not access the negotiated host memory or send an outbound IOMB. Any SAS or SATA I/O related DMA operation that has been initiated will not be aborted and will continue until that portion of the DMA operation is completed. Therefore, the host must make sure that the I/O operation is quiesced before terminating the MPI communication.

To terminate the host and the SPC 8x6G-negotiated messaging configuration table:

1. The host notifies the SPC 8x6G controller by setting bit 1 of the [Inbound Doorbell Register](#) as described in Section [10.2.1](#).
2. The SPC 8x6G MSGU terminates the host and the SPC 8x6G-negotiated messaging configuration table.
3. The host reads back the [Inbound Doorbell Register](#) (by polling) until both register bit 1 and the MPI-S field in DWord 0, bit [2:0], of the General Status Table (GST) described in [Table 39](#) is cleared. This indicates that the SPC 8x6G controller has acknowledged the termination of the negotiated configuration table.

5.2.6.3 Host-SPC 8x6G MPI Inbound Freeze

This section describes the sequence to freeze MPI inbound communication between a host and the SPC 8x6G that was established at the completion of the sequence described in Section [5.2.6.1](#), “[Host-SPC 8x6G MPI Initialization](#)” above.

This MPI freeze operation is only applicable to the SPC 8x6G MSGU firmware unit for the inbound direction. As soon as the SPC 8x6G receives the freeze notification, it will stop fetching new inbound IOMBs from a particular IQ until it receives an un-freeze notification.

The IQ freeze may be used for the following purposes:

- To stop the SPC 8x6G from fetching new IOMBs from the IQ when the host intends to roll back the IQ Producer Index.
- To stop the SPC 8x6G from fetching new IOMBs from the IQ when the host intends to invalidate some of the IOMBs that have been queued (but not fetched by the SPC 8x6G) in the particular host Inbound Queue. Invalidation of IOMB is done by clearing the V bit in the IOMB header.

The freeze operation may be applied on a per IQ basis. The [Host Scratchpad 1 Register](#) and the [Inbound Doorbell Register](#) are used to indicate which IQs to be frozen. Bit 0 in the [Host Scratchpad 1 Register](#) represents IQ 0 and bit 31 in the [Host Scratchpad 1 Register](#) represents IQ 31. Bit 0 in the [Host Scratchpad 2 Register](#) represents IQ 32 and bit 31 in the [Host Scratchpad 2 Register](#) represents IQ 63. (See Sections [10.2.10](#) [10.2.11](#) respectively for definitions of these registers.)

Multiple consecutive freeze operations are allowed by setting the IQ number to be frozen to one in the [Host Scratchpad 1 Register](#) and the [Host Scratchpad 2 Register](#). A bit setting of zero is ignored and does not unfreeze the IQ.

The freeze operation does not impact the inbound IOMB requests that are already fetched by the SPC 8x6G or outbound IOMB operations.

To freeze the host and the SPC 8x6G inbound IOMB messaging:

1. The host sets the corresponding bit(s) in the [Host Scratchpad 1 Register](#) and the [Host Scratchpad 2 Register](#) to indicate which IQ(s) to freeze. Clear the bit(s) corresponding to the IQ(s) that are to be left unchanged.
2. The host notifies the SPC 8x6G controller by setting bit 2 of the [Inbound Doorbell Register](#) as described in Section [10.2.1](#).
3. The SPC 8x6G MSGU stops fetching new inbound IOMB entries from the corresponding IQ(s) specified in step (1) above.
4. The host reads back the [Inbound Doorbell Register](#) (by polling up to 10 microseconds) until the register bit 2 is cleared. This indicates that the SPC 8x6G controller has acknowledged the freezing of the corresponding inbound IOMB processing.

5.2.6.4 Host-SPC 8x6G MPI Inbound Un-freeze

This section describes the sequence to un-freeze MPI inbound communication between a host and the SPC 8x6G that was established at the completion of the sequence described in Section 5.2.6.3, “Host-SPC 8x6G MPI Inbound Freeze” above.

The [Host Scratchpad 1 Register](#) and the [Host Scratchpad 2 Register](#) are used to indicate which IQ(s) to un-freeze. Bit 0 in the [Host Scratchpad 1 Register](#) represents IQ 0 and bit 31 in the [Host Scratchpad 1 Register](#) represents IQ 31. Bit 0 in the [Host Scratchpad 2 Register](#) represents IQ 32 and bit 31 in the [Host Scratchpad 2 Register](#) represents IQ 63. (See Sections 10.2.10 10.2.11 respectively for definitions of these registers.)

To un-freeze the host and the SPC 8x6G inbound IOMB messaging:

1. The host sets the corresponding bit(s) in the [Host Scratchpad 1 Register](#) and [Host Scratchpad 2](#) to indicate which IQ(s) to un-freeze. Clear the bit(s) corresponding to the IQ(s) that are to be left unchanged.
2. The host notifies the SPC 8x6G controller by setting bit 3 of the [Inbound Doorbell Register](#) as described in Section [10.2.1](#).
3. The SPC 8x6G MSGU resumes fetching new inbound IOMB entries in the corresponding IQ.
4. The host reads back the [Inbound Doorbell Register](#) (by polling) until the register bit 3 is cleared. This indicates that the SPC 8x6G controller has acknowledged the un-freezing of the inbound IOMB processing.

6 Common IOMB Header

The IOMB structure consists of a common header section (4 bytes) and an operation-specific Message Payload.

Figure 48 IOMB Command Header

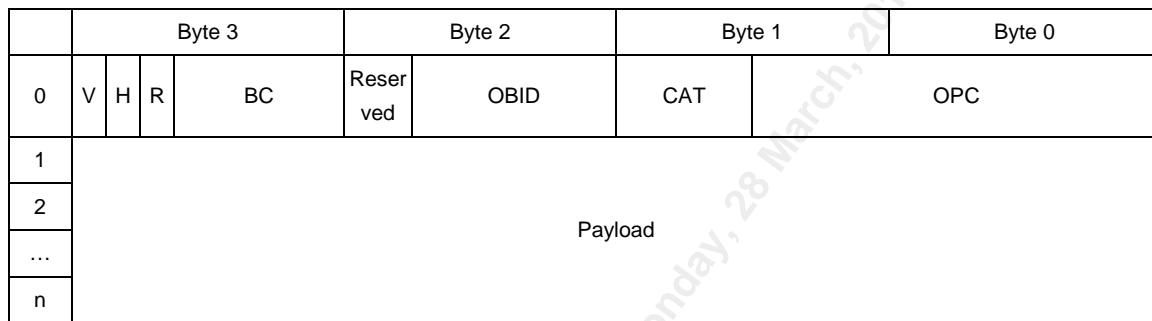


Figure 48 shows the general message format. The fields in the header are defined in Table 43.

Table 43 IOMB Command Header Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
0 [31]	Valid	V	1 bit	<p>Valid bit.</p> <p><i>For an inbound IOMB:</i></p> <p>If the host posts an IOMB with the valid (V) bit not set or the host clears the V bit after the IOMB is posted but before the SPC 8x6G fetches it, the SPC 8x6G will not execute the IOMB as it normally would, but will explicitly return the IOMB to the OQ through a GENERAL_EVENT Notification. (See Section 8.18.)</p> <p>After the IOMB is fetched, the SPC 8x6G increments the queue's Consumer Index (CI). The SPC 8x6G does not clear the V bit.</p> <p><i>For an outbound IOMB:</i></p> <p>If the valid bit is not set, the host discards the IOMB and increments the Outbound Consumer Index.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
0 [30]	High Priority	H	1 bit	<p>High priority bit. Set for a high priority IOMB and cleared for a normal priority IOMB.</p> <p><i>For an inbound IOMB:</i></p> <p>To indicate to the SPC 8x6G that this IOMB is a high priority IOMB sent to the high priority inbound queue. This high priority flag is required in the case where IOMBs are queued inside the SPC 8x6G. When IOMBs are queued inside SPC 8x6G, the priority information of the Inbound Queue from which this IOMB is received is not saved. SPC 8x6G will only use this H bit in the IOMB header information to get the priority information.</p> <p><i>For an outbound IOMB:</i></p> <p>To indicate to the host that the outbound IOMB is a response to a high priority inbound IOMB.</p>
0 [28:24]	Buffer Count	BC	5 bits	<p>Buffer Count (5 bits). Number of consecutive entries in the circular queue used by the message. This field is primarily used for outbound messages since the inbound message Buffer Count is always 1.</p> <p>Messages can span across several circular queue entries when the message size is bigger than the circular queue entry size. In these cases, the message uses logically consecutive entries in the queue.</p> <p>SPC 8x6G-specific use: When this field is set to a value larger than 1, the IOMB header is not repeated for the second and subsequent blocks.</p>
0 [21:16]	Outbound Queue ID	OBID	6 bits	<p>OQ ID. Set when sending inbound message to specify which OQ (in a multi-OQ configuration) to use for outbound completion.</p> <p>The host is responsible for setting the correct value as the host is responsible for determining the OQ where the response is returned.</p>
0 [15:12]	Category	CAT	4 bits	<p>This field carries the system-defined, system-global value that indicates which function (unit hardware and driver software instance) is associated with this message:</p> <ul style="list-style-type: none"> 0x00 = iSCSI/Ethernet 0x01 = Fibre Channel 0x02 = SAS/SATA 0x03 = SCSI <p>For the SPC 8x6G, only the values 0x02 (SAS/SATA) and 0x03 (SCSI) are valid.</p>
0 [11:0]	Operation Code	OPC	12 bits	The specific operation code for the IOMB.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
n:1	Payload		n * 4 bytes	This field is operation-dependent.

7 Inbound Messages

Table 44 summarizes the inbound operation codes.

Table 44 Summary of Inbound Operation Codes

Op Code (Hex)	Command/Mnemonic	Usage Initiator/Target	IOMB Size (Bytes)	Section
0x0001	ECHO Command	I/T	64	7.1
0x0004	PHY_START Command	I/T	64	7.2
0x0005	PHY_STOP Command	I/T	64	7.3
0x0006	SSP_INI_IO_START Command	I	64	7.4
0x0007	SSP_INI_TM_START Command	I	64	7.5
0x0008	SSP_INI_EXT_IO_START Command	I	>64	7.6
0x0009	DEVICE_HANDLE_ACCEPT Command	T	64	7.7
0x000A	SSP_TGT_IO_START Command	T	64	7.8
0x000B	SSP_TGT_RESPONSE_START Command	T	64	7.9
0x000F	SSP_ABORT Command	I/T	64	7.10
0x0010	DREGISTER_DEVICE_HANDLE Command	I/T	64	7.11
0x0011	GET_DEVICE_HANDLE Command	I/T	64	7.12
0x0012	SMP_REQUEST Command	I	64	7.13
0x0014	SMP_ABORT Command	I/T	64	7.14
0x0016	REGISTER_DEVICE Command	I	64	7.15
0x0017	SATA_HOST_IO_START Command	I	64	7.16
0x0018	SATA_ABORT Command	I	64	7.17
0x0019	LOCAL_PHY_CONTROL Command	I/T	64	7.18
0x001A	GET_DEVICE_INFO Command	I/T	64	7.19
0x0020	FW_FLASH_UPDATE Command	I/T	64	7.20
0x0022	GPIO Command	I/T	64	7.21
0x0023	SAS_DIAG_MODE_START_END Command	I/T	64	7.22
0x0024	SAS_DIAG_EXECUTE Command	I/T	64	7.23
0x0025	SAS_HW_EVENT_ACK Command	I/T	64	7.24
0x0026	GET_TIME_STAMP Command	I/T	64	7.25
0x0027	PORT_CONTROL Command	I/T	64	7.26
0x0028	GET_NVMD_DATA Command	I/T	64	7.27
0x0029	SET_NVMD_DATA Command	I/T	64	7.28
0x002A	SET_DEVICE_STATE Command	I/T	64	7.29
0x002B	GET_DEVICE_STATE Command	I/T	64	7.30
0x002C	SET_DEVICE_INFO Command	I/T	64	7.31

Op Code (Hex)	Command/Mnemonic	Usage Initiator/Target	IOMB Size (Bytes)	Section
0x002D	SAS_RE_INITIALIZATION Command	I/T	64	7.32
0x002E	SGPIO_REGISTER Command	I	64	7.33
0x002F	PCI_DIAG_EXECUTE Command	I/T	64	7.34

7.1 ECHO Command

Description

This command requests an echo with the specified payload from the SPC 8x6G controller. This command tests the MPI between the host and the SPC 8x6G controller for proper operation.

The command's response is reported in the [ECHO Response](#). (See Section 8.1.)

Usage

Initiator and target.

Command Format

Table 45 ECHO Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x001
1	HTAG							
2	PAYLOAD							
4	PAYLOAD							
...								
15								

Table 46 ECHO Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Payload	PAYLOAD	56 Bytes	Host-defined payload.

7.2 PHY_START Command

Description

This command is sent to initialize and bring the SAS link up on the corresponding PHY.

The host is notified of the progress of the PHY_START command via the [SAS_HW_EVENT Notification](#) as described in Section 8.2:

- The first notification is sent via the [SAS_HW_EVENT Notification](#) with the event code, IOP_EVENT_PHY_START_STATUS, to indicate the status of the operation.
- Then if the PHY is connected to another device at the other side of the link, the host is notified again when the PHY goes to a link up state via the [SAS_HW_EVENT Notification](#) with the event code IOP_EVENT_SAS_PHY_UP or IOP_EVENT_SATA_PHY_UP.

See Section 4.12.1, “[PHY Initialization](#)” for the PHY initialization sequence.

The OBID for this IOMB is ignored and the default OQ for the SAS_HW_EVENT Notification is used based on the setting of the Configuration Table DWord 0x0A (Phy 0 to 3) and 0x0B (Phy 4 to 7) described in in [Table 38](#).

To configure a wide port, a set of PHYs is individually started using this IOMB command with the same SAS Identify Address Frame (SASIDAF).

If a PHY is configured as part of a wide port, all PHYs belonging to the same wide port must be configured with the same properties as specified by the Spin Up Hold (SH), Link Mode (LM), and Supported Link Rate (SLR) fields.

The host needs to manage the spinning-up of a directly-attached SAS drive either using the SCSI START UNIT command or the TX NOTIFY SPINUP operation of the [LOCAL_PHY_CONTROL Command](#). (See Section 7.18.)

Usage

Initiator and target.

IOMB Format

Table 47 PHY_START Command Format

	Byte 3				Byte 2		Byte 1			Byte 0			
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x004					
1	HTAG												
2	Reserved				SSCD	AS E	SH	LM	SLR	Reserved	PHYID		
3	SASIDAF												
9													
10										SPASTI			
11	Reserved												
...													
15													

Table 48 PHY_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [18:16]	SSC Disable	SSCD	4 bit	This bit disables the SSC for a particular speed. 0001b : Disable 1.5 Gbits/sec SSC 0010b : Disable 3.0 Gbits/sec SSC 0100b : Disable 6.0 Gbits/sec SSC Note: This field is only valid if the SLR 6GBits/sec is set.
2 [15]	SAS Analog Setup Enable	ASE	1 bit	This bit enables the option to change the PHY analog setup. See Section 5.2.5, “MPI SAS PHY Analog Setup Table Fields”. 0b: Do not change current PHY analog setup. 1b: Enable the use of the SAS PHY Analog Setup Table. The SPASTI field will be used as an index to retrieve the values from Table 42, “SAS PHY Analog Setup Table (SPAST)”.
2 [14]	Enable Spin Up Hold	SH	1 bit	Controls the spin up hold of the directly attached SATA device: 0b: Disable spin up hold 1b: Enable spin up hold

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2 [13:12]	Link Mode	LM	2 bits	Controls whether SAS and/or direct attached SATA mode is enabled: 01b: SAS mode (SSP, SMP and STP) 10b: Direct-attached SATA mode 11b: Auto mode. Supports both SAS (SSP, SMP and STP) and direct-attached SATA mode
2 [11:8]	Supported Link Rate	SLR	4 bits	The PHY supported link rate. More than one bit can be set: 0001b: 1.5 Gbit/s 0010b: 3 Gbit/s 0100b: 6 Gbit/s
2 [3:0]	PHY Identifier	PHYID	4 bits	Zero-based PHY Identifier.
[9:3]	SAS Identify Address Frame	SASIDAF	28 Bytes	The SAS Identify Address Frame sent from the local PHY, excluding the last 4 bytes of the CRC. Big Endian format. See Section 2.1.3.1, “ SAS Payload Endianness ” for more details about payload endianness.
10	SAS PHY Analog Setup Table Index	SPASTI	1 Byte	Valid only if ASE is set to 1. Index to SAS PHY Analog Setup Table. See Section 5.2.5, “ MPI SAS PHY Analog Setup Table Fields ”.

7.3 PHY_STOP Command

Description

This command is sent to an instance of the PHY in order to set it into link down state. This command must only be used once the PHY has been initialized via the [PHY_START Command](#) described in Section 7.2.

The host is informed when the PHY instance has properly stopped (entered the link down state) via the [SAS_HW_EVENT Notification](#) message with an event code of IOP_EVENT_PHY_DOWN. See Section 8.2. The OBID for this IOMB is ignored and the default OQ for the [SAS_HW_EVENT Notification](#) is used based on the setting of Configuration Table DWord 0x09 [7:0] described in [Table 38](#).

Usage

Initiator and target.

Command Format

Table 49 PHY_STOP Command Format

	Byte 3				Byte 2		Byte 1		Byte 0					
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x005						
1	HTAG													
2	Reserved				Reserved				Reserved	PHYID				
3	Reserved													
...														
15														

Table 50 PHY_STOP Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [3:0]	PHY Identifier	PHYID	4 bits	Zero-based PHY Identifier.

7.4 SSP_INI_IO_START Command

Description

The SSP Initiator I/O Start command initiates the SSP command with a CDB length of up to 16 bytes.

The SPC 8x6G controller allocates an internal I/O context and sends an SSP_COMMAND frame to the designated target device. See Section 3.5, “[SSP Initiator Write Operations](#)” and Section 3.6, “[SSP Initiator Read Operations](#)”.

The SSP_RESPONSE frame sent by the target device is reported in the [SSP_COMPLETION Response](#) as described in Section 8.3.

Usage

Initiator.

Command Format

Table 51 SSP_INI_IO_START Command Format

	Byte 3			Byte 2		Byte 1		Byte 0			
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02	OPC=0x006		
1	HTAG										
2	DEVICE_ID										
3	DL										
4	Reserved						DIR	Reserved	M TLR		
5	SSPIU (First 28 Bytes)										
...											
11											
12											
13	SGLAL										
14	SGLAH										
15	E	LEN									
		Reserved									

Table 52 SSP_INI_IO_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation. This HTAG is only to be used to communicate between a host and the SPC 8x6G. The SPC 8x6G firmware assigns the SSP frame header and its SAS TAG field.
2	Device Identifier	DEVICE_ID	4 Bytes	SSP target device identifier. See Section 3.2 for a detailed description of DEVICE_ID.
3	Data Length	DL	4 Bytes	Expected data transfer length.
4 [9:8]	Direction	DIR	2 bits	Direction of data transfer: 00b: No data transfer 01b: Inbound from the SAS/SATA link. Data transfer from the target to the initiator (the SPC 8x6G) to the host. 10b: Outbound to the SAS/SATA link. Data transfer from the host/initiator (SPC 8x6G) to the target.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [2]	Message Report	M	1 bit	<p>This bit provides the option for the host to ask the SPC 8x6G to send an event to the host when the frame has been sent on the wire.</p> <p>The event IO_XFER_CMD_FRAME_ISSUED in the SSP_EVENT Notification is used to notify the host that the frame has been sent and an ACK primitive has been received. See Section 8.11.</p> <p>1b: Send the event after the frame is sent. 0b: Do not send the event after the frame is sent. This is the default setting.</p>
4 [1:0]	TLR CONTROL	TLR	2 bits	<p>TLR control field:</p> <p>00b: Enable TLR based on target mode page. Use for both SAS 1.1 TLR and SAS 2.0 TLR.</p> <p>01b: Enable TLR on per-command basis</p> <p>10b: Disable TLR</p> <p>11b: Enable TLR based on target mode page. Use for SAS 2.0 TLR only.</p> <p>For details, see Section 3.20, “Transport Layer Retry (TLR) Handling”.</p>
[11:5]	SSP Information Unit	SSPIU	28 Bytes	<p>The first 28 bytes of an SSPIU excluding the ADDITIONAL_CDB_BYTES.</p> <p>The ADDITIONAL_CDB_LENGTH should be set to zero.</p> <p>Big Endian format. See Section 2.1.3.1, “SAS Payload Endianness” for details about the payload endianness.</p>
12	SGL Address Low	SGLAL	4 Bytes	<p>If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory.</p> <p>If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory.</p>
13	SGL Address High	SGLAH	4 Bytes	<p>If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory.</p> <p>If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory.</p>
14	Length	LEN	4 Bytes	<p>If Local SGL is used, this field contains the size in bytes of the data buffer in host memory.</p> <p>If ESGL is used, this field is not used.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
15 [31]	Extension bit	E	1 bit	This bit indicates if this SGL element (L/A pair) describes a data buffer or an ESGL in host memory: 0b: This SGL element describes a data buffer 1b: This SGL element describes an ESGL

7.5 SSP_INI_TM_START Command

Description

The SSP Initiator Task Management Start command is sent to initiate an SSP task management operation into the designated target device.

This command can also abort an SSP I/O when the firmware has fetched the SSP I/O but has not yet processed it. The SSP Completion Status field in the SSP_COMPETITION Response shows the status as IO_ABORTED with a bit set in the PARAM field. It indicates that the SSP I/O was aborted by an SSP_INI_TM_START command.

The response from the target device is reported in the [SSP_COMPLETION Response](#) as described in Section [8.3](#).

Usage

Initiator.

Command Format

Table 53 SSP_INI_TM_START Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x007
1	HTAG							
2	DEVICE_ID							
3	HTAG Association (HTAG-A)							
4	Task Management Function (TMF)							
5	LUN							
6								
7	Reserved							D S A D S M Reser ved
8	Reserved							
...								
15								

Table 54 SSP_INI_TM_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	SSP target device identifier. See Section 3.2, “Device Handle and DEVICE_ID” for a detailed description of DEVICE_ID.
3	HTAG Association	HTAG-A	4 Bytes	<p>HTAG of the task to be managed. Only valid if TMF is set to 0x01 for ABORT_TASK or 0x80 for QUERY_TASK.</p> <p>This is the HTAG that was previously passed to the SSP_INI_IO_START Command described in Section 7.4. Based on this HTAG-A, the firmware assigns the proper associated TAG OF TASK TO BE MANAGED field in the Task Management function SSP IU.</p> <p>If the HTAG-A passed is invalid (the SPC 8x6G could not find the matching tag), this task management request will be returned with the status IO_TM_TAG_NOT_FOUND in the SSP_COMPLETION Response described in Section 8.3.</p>
4	Task Management Function	TMF	4 Bytes	<p>Task management function:</p> <ul style="list-style-type: none"> 0x01: ABORT_TASK 0x02: ABORT_TASK_SET 0x04: CLEAR_TASK_SET 0x08: LOGICAL_UNIT_RESET 0x10: I_T_NEXUS_RESET 0x20: Reserved 0x40: CLEAR ACA 0x80: QUERY_TASK 0x81: QUERY_TASK_SET 0x82: QUERY ASYNCHRONOUS EVENT
[6:5]	Logical Unit Number	LUN	8 Bytes	<p>SCSI logical unit number.</p> <p>Big Endian format. See Section 2.1.3.1, “SAS Payload Endianness” for details about SAS payload endianness.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
7 [4]	Set Device State In Recovery Flag	DS	1 bit	<p>An option for the host to set the Device State to DS_IN_RECOVERY, which will block and flush (return with error) all I/O requests destined to the target device except for this task management request. This option is applicable for all types of task management functions. For the ABORT_TASK function, the SPC 8x6G will set the Device State to DS_IN_RECOVERY only if the SPC 8x6G finds the matching HTAG-A.</p> <p>The ADS field (see below) provides an additional option for an ABORT_TASK function operation when the SPC 8x6G cannot find the matching HTAG-A.</p> <p>When the device state is set to DS_IN_RECOVERY, all I/O requests (except this task management request) to the device specified by that DEVICE_ID will be returned with a STATUS_IO_DS_IN_RECOVERY error code.</p> <p>See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states.</p> <p>0b: Do not change the device state. 1b: Set the device state to DS_IN_RECOVERY.</p>
7 [3]	Abort Task option for Set Device State In Recovery Flag	ADS	1 bit	<p>This bit is valid only when the TMF bit is set to ABORT_TASK(0x01):</p> <p>0b: Do not set the device state to DS_IN_RECOVERY if the SPC 8x6G cannot find the matching HTAG-A.</p> <p>1b: Set the device state to DS_IN_RECOVERY if the SPC 8x6G cannot find the matching HTAG-A.</p>
7 [2]	Message Report	M	1 bit	<p>This bit provides the option for the host to ask the SPC 8x6G to send an event to the host when the frame has been sent on the wire.</p> <p>The event IO_XFER_CMD_FRAME_ISSUED in the SSP_EVENT Notification is used to notify the host that the frame has been sent and an ACK primitive has been received. See Section 8.11.</p> <p>1b: Send the event after the frame is sent. 0b: Do not send event after the frame is sent. This is the default.</p>

7.6 SSP_INI_EXT_IO_START Command

Description

The SSP Initiator Extended I/O Start command is sent to initiate the SSP command with a CDB length of more than 16 bytes.

The SPC 8x6G controller allocates an internal I/O context and sends an SSP_COMMAND frame to the designated target device. See Section 3.5, “[SSP Initiator Write Operations](#)” and Section 3.6, “[SSP Initiator Read Operations](#)”.

The SSP_RESPONSE frame sent by the target device is reported in the [SSP_COMPLETION Response](#) message described in Section 8.3.

Usage

Initiator.

Command Format

Table 55 SSP_INI_EXT_IO_START Command Format

	Byte 3			Byte 2		Byte 1		Byte 0			
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02	OPC=0x008		
1	HTAG										
2	DEVICE_ID										
3	DL										
4	SSPIUL				Reserved		DIR	Reserved	M TLR		
5	SSPIU										
...											
M+5											
M+6	SGLAL										
M+7	SGLAH										
M+8	LEN										
M+9	E0	Reserved									

Table 56 SSP_INI_EXT_IO_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	Device Identifier	DEVICE_ID	4 Bytes	SSP target device identifier. See Section 3.2, “ Device Handle and DEVICE_ID ” for a detailed description of DEVICE_ID.
3	Data Length	DL	4 Bytes	Expected data transfer length.
4 [31:16]	SSP Information Unit Length	SSPIUL	2 Bytes	Length in bytes of the SSPIU.
4 [9:8]	Direction	DIR	2 bits	Direction of data transfer: 00b: No data transfer 01b: Inbound from the SAS/SATA link. Data transfer from the target to the initiator (SPC 8x6G) to the host 10b: Outbound to the SAS/SATA link. Data transfer from the host/initiator (SPC 8x6G) to the target.
4 [2]	Message Report	M	1 bit	This bit provides the option for host to ask SPC 8x6G to send an event to host when the frame has been sent on the wire. The event IO_XFER_CMD_FRAME_ISSUED in the SSP_EVENT Notification is used to notify the host that the frame has been sent and an ACK primitive has been received. See Section 8.11. 1b: Send the event after the frame is sent. 0b: Do not send event after the frame is sent. This is the default setting.
4 [1:0]	TLR CONTROL	TLR	2 bits	TLR control field: 00b: Enable TLR based on target mode page. Use for both SAS 1.1 TLR and SAS 2.0 TLR. 01b: Enable TLR on per-command basis 10b: Disable TLR 11b: Enable TLR based on target mode page. Use for SAS 2.0 TLR only. For details, see Section 3.20, “ Transport Layer Retry (TLR) Handling ”.
[11:5]	SSP Information Unit	SSPIU	M Bytes	The SSP information unit including the ADDITIONAL_CDB_BYTES. The ADDITIONAL_CDB_LENGTH contains the length in DWords (four bytes) of the ADDITIONAL CDB field and should be set appropriately. The current maximum supported ADDITIONAL_CDB_LENGTH is five DWords (20 bytes) for the maximum CDB length of 36 bytes. Big Endian format. See Section 2.1.3.1, “ SAS Payload Endianness ” for details about SAS payload endianness.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
M+6	SGL Address Low	SGLAL	4 Bytes	If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory.
M+7	SGL Address High	SGLAH	4 Bytes	If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory.
M+8	Length	LEN	4 Bytes	If Local SGL is used, this field contains the size in bytes of the data buffer in host memory. If ESGL is used, this field is not used.
M+9 [31]	Extension bit	E	1 bit	This bit indicates if this SGL element (L/A pair) describes a data buffer or an ESGL in host memory: 0b: This SGL element describes a data buffer 1b: This SGL element describes an ESGL

7.7 DEVICE_HANDLE_ACCEPT Command

Description

In target mode, when the SPC 8x6G controller receives a command from a remote initiator device, it searches for the device in the internal device context table. If the device is not found, the controller reports the new device by sending a [DEVICE_HANDLE_ARRIVED](#) Notification to the host. See Section 8.12.

The host uses this command in order to accept or reject the creation of the new internal context for this device:

- If the host sets the DEVA field to ACCEPT device action (0x00), the SPC 8x6G controller processes the received command.
- If the host sets the DEVA field to REJECT device action (0x01), the SPC 8x6G controller drops the received command.

Usage

Target.

Command Format

Table 57 DEVICE_HANDLE_ACCEPT Command Format

	Byte 3			Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02 OPC=0x009				
1	HTAG											
2	CTAG											
3	INITIATOR DEVICE_ID											
4	DEVA		Reserved		A	HA	IT Nexus Timeout (ITNT)					
5	Reserved											
...												
15												

Table 58 DEVICE_HANDLE_ACCEPT Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	SPC 8x6G Controller Tag	CTAG	4 Bytes	The SPC 8x6G Controller Tag or context received in the DEVICE_HANDLE_ARRIVED Notification . See Section 8.12.
3	DEVICE_ID	DEVICE Identifier	4 Bytes	The SSP Initiator device identifier received in the DEVICE_HANDLE_ARRIVED Notification . See Section 8.12. See Section 3.2 for a detailed description of DEVICE_ID.
4 [31:24]	Device Action	DEVA	1 Byte	Device action code: 0x00: ACCEPT device 0x01: REJECT device
4 [17]	AWT flag	A	1 bit	Priority setting for the Arbitration Wait Time (AWT) for this initiator: 0b: Default setting (recommended). The actual AWT value is based on how long an OPEN frame has been waiting for a connection request to be accepted. It starts at 0. As specified in the SAS specification, from 0 to 32768 µs, the AWT value is incremented every µs. From 32768 µs and on, the AWT value is incremented every ms. 1b: Increase priority. The actual AWT value starts at 32768 µs when the A flag is set to '1'. The AWT value is incremented every ms from that point on.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [16]	Host Assigned Upper DEVICE_ID	HA	1 bit	When set, this indicates that the host will assign the upper 16-bit of the DEVICE_ID field. The host may assign bits [15:0] for its own purposes. When cleared, the SPC 8x6G 32-bit assigned DEVICE_ID will be used.
4 [15:0]	IT Nexus Timeout	ITNT	2 Bytes	The value in milliseconds of the time unit that is used by the SPC 8x6G to determine the nexus timeout condition.

7.8 SSP_TGT_IO_START Command

Description

The host uses the SSP Target I/O Start command to:

- Send an SSP XFER_RDY frame for a Write I/O.
- Initiate a data transfer for a Read I/O

The SPC 8x6G controller allocates an internal I/O context and does the requested operation. See Section 3.7, “SSP Target Write Operations” and Section 3.8, “SSP Target Read Operations” for a full description.

The completion response for this command is reported in the [SSP_COMPLETION Response](#) as described in Section 8.3.

Usage

Target.

Command Format

Table 59 SSP_TGT_IO_START Command Format

	Byte 3				Byte 2		Byte 1			Byte 0							
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x00A									
1	HTAG																
2	DEVICE_ID																
3	DL																
4	DO																
5	INI_TAG						O D S	Reserved		DIR	A G R	RD F	RTE	AN	Reser ved		
6	Reserved																
...																	
11																	
12	SGLAL																
13	SGLAH																
14	LEN																
15	E	Reserved															

Table 60 SSP_TGT_IO_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Host tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	SSP target device identifier. See Section 3.2, "Device Handle and DEVICE_ID" for a detailed description of DEVICE_ID.
3	Data Length	DL	4 Bytes	Expected data transfer length.
4	Data Offset	DO	4 Bytes	This is the offset into the overall data block where this data phase is to begin. If this I/O uses a single data phase, then the offset is 0.
5 [31:16]	Initiator Tag	INI_TAG	2 Bytes	SSP TAG received in the SSP_REQUEST_RECEIVED Notification . See Section 8.13.
5 [15]	Override Device State	ODS	1 bit	Indicates firmware should transmit the Target IO even when the DEVICE_STATE is DS_IN_RECOVERY: 0b: Do not override device state 1b: Override device state

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
5 [9:8]	Direction	DIR	2 bits	<p>Direction of data transfer:</p> <p>00b: No data transfer.</p> <p>01b: Inbound from the SAS/SATA link. Data transfer from remote initiator to target (SPC 8x6G) to the host.</p> <p>10b: Outbound to the SAS/SATA link. Data transfer from the host/target (SPC 8x6G) to the remote initiator.</p> <p>For an operation that does not involve a data transfer, this field is ignored.</p>
5 [7]	Auto Good Response	AGR	1 bit	Auto good response on a successful read (data transfer from target to initiator) request.
5 [6]	Retry Data Frame	RDF	1 bit	<p>Retry Data Frame.</p> <p>Indicates whether the target supports the RTL for this particular I/O. This bit MUST be set if the RTE and/or AN fields are set to enable TLR.</p>
5 [5:4]	TLR support for data phases	RTE	2 bit	<p>TLR encoding for data phase:</p> <p>00b: No retry</p> <p>01b: Retry on ACK/NAK timeout</p> <p>10b: Retry on NAK received</p> <p>11b: Retry on both ACK/NAK timeout and NAK received</p> <p>For details, see Section 3.20, “Transport Layer Retry (TLR) Handling”.</p> <p>If the RTE field is set to enable TLR, the RDF field must be set.</p>
5 [3:2]	TLR support for other phases	AN	2 bit	<p>TLR encoding for other phases:</p> <p>00b: No retry</p> <p>01b: Retry on ACK/NAK timeout</p> <p>10b: Retry on NAK received</p> <p>11b: Retry on both ACK/NAK timeout and NAK received.</p> <p>For details, see Section 3.20, “Transport Layer Retry (TLR) Handling”.</p> <p>If the AN field is set to enable TLR, the RDF field must be set.</p>
12	SGL Address Low	SGLAL	4 Bytes	<p>If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory.</p> <p>If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
13	SGL Address High	SGLAH	4 Bytes	If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory.
14	Length	LEN	4 Bytes	If Local SGL is used, this field contains the size in bytes of the data buffer in host memory. If ESGL is used, this field is not used.
15 [31]	Extension bit	E	1 bit	This bit indicates if this SGL element (L/A pair) describes a data buffer or an ESGL in host memory: 0b: This SGL element describes a data buffer 1b: This SGL element describes an ESGL

7.9 SSP_TGT_RESPONSE_START Command

Description

The host sends SSP Target Response Start command in order to send an SSP_RESPONSE frame to the specified target device.

The completion notification for this command is reported in the [SSP_COMPLETION Response](#) as described in Section [8.3](#).

Usage

Target.

Command Format

Table 61 SSP_TGT_RESPONSE_START Command Format

	Byte 3				Byte 2		Byte 1			Byte 0							
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x00b									
1	HTAG																
2	DEVICE_ID																
3	RL																
4	INI_TAG					ODS	Reserved	D	Reserved	AN	Reserved						
5	Reserved or SSP_RESPONSE Frame (SSPRF[27:0])																
...																	
11																	
12																	
13																	
14	SGLAL or SSPRF[31:28]																
15	SGLAH or SSPRF[35:32]																
	LEN or SSPRF[39:36]																
	Reserved or SSPRF[43:40]																

Table 62 SSP_TGT_RESPONSE_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	The SSP initiator device identifier. See Section 3.2 for a detailed description of DEVICE_ID.
3	Response Length	RL	4 Bytes	Total response length in bytes.
4 [31:16]	Initiator Tag	INI_TAG	2 Bytes	The SSP TAG received in the SSP_REQUEST_RECEIVED Notification . See Section 8.13.
4 [15]	Override Device State	ODS	1 bit	Indicates firmware should transmit the RESPONSE even when the DEVICE_STATE is non-operational: 0b: Do not override device state 1b: Override device state

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [10]	Direct Payload	D	1 bit	Indicates the location of the SSP_RESPONSE frame to send: 0b: Use DWord [15:12] as the physical address in the host memory where the SSP_RESPONSE frame to send resides. 1b: Use the inbound IOMB for the SSP_RESPONSE frame. Only for SSP_RESPONSE frames up to 44 bytes long. DWord [11:5] contains the first 28 bytes of the SSP_RESPONSE frame.
4 [3:2]	ACK NAK Retry	AN	2 bit	ACK and NAK retry option: 00b: No retry 01b: Retry on ACK/NAK timeout 10b: Retry on NAK received 11b: Retry on both ACK/NAK timeout and NAK received
5	SSP_RESPONSE Frame	SSPRF	28 Bytes	The first 28 bytes of the SSP_RESPONSE frame to send. Valid only if the D flag is set to 1b. Big Endian format. See Section 2.1.3.1, "SAS Payload Endianness" for details about SAS payload endianness.
12	SGL Address Low or SSPRF[31:28]	SGLAL or SSPRF[31:28]	4 Bytes	If the D flag is set to 0b, this field contains the lower 32-bit physical address, in host memory, of the buffer that contains the SSP_RESPONSE frame. <ul style="list-style-type: none"> • If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory. • If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory. If the D flag is set to 1b this field contains the SSP_RESPONSE frame bytes 28 to 31.
13	SGL Address High or SSPRF[35:32]	SGLAH or SSPRF[35:32]	4 Bytes	If the D flag is set to 0b, this field contains the higher 32-bit physical address, in host memory, of the buffer that contains the SSP_RESPONSE frame. <ul style="list-style-type: none"> • If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory. • If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory. If the D flag is set to 1b this field contains the SSP_RESPONSE frame bytes 32 to 35.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
14	Length or SSPRF[39:36]	LEN or SSPRF[39:36]	4 Bytes	If the D flag is set to 0b, this field contains the size of the SGL/ESGL: <ul style="list-style-type: none">• If Local SGL is used, this field contains the size in bytes of the data buffer in host memory.• If ESGL is used, this field is not used. If the D flag is set to 1b this field contains the SSP_RESPONSE frame bytes 36 to 39.
15	Reserved or SSPRF[43:40]	Reserved or SSPRF[43:40]	4 Bytes	If the D flag is set to 0b, this field is reserved. If the D flag is set to 1b, this field contains the SSP_RESPONSE frame bytes 40 to 43.

7.10 SSP_ABORT Command

Description

This command is sent to abort one or more I/O request previously initiated by any of the following commands:

- [SSP_INI_IO_START Command](#) (Section 7.4)
- [SSP_INI_TM_START Command](#) (Section 7.5)
- [SSP_INI_EXT_IO_START Command](#) (Section 7.6)
- [SSP_TGT_IO_START Command](#) (Section 7.8)
- [SSP_TGT_RESPONSE_START Command](#) (Section 7.9)

The SSP_ABORT command only aborts or invalidates pending I/O(s) and their associated internal resources in the SPC 8x6G controller. It does not send any protocol specific command to abort an I/O operation in the target device.

The SSP_ABORT command has an option field (SCP) to indicate the scope of abort: to abort an individual I/O with a specific HTAG-ABT or to abort all I/Os associated with a DEVICE_ID.

If the NOQ bit is not set, the SPC 8x6G controller releases I/O resources for the aborted SSP I/O after a delay which is two times the IT Nexus timeout for the device.

Table 63 SSP_ABORT Scope of Abort (SCP)

Option	SCP Encoding	Description
Aborting a Single SSP I/O	00b	<p>Typically this command is sent after the completion of a successful task management command to abort the corresponding I/Os in target device. See Section 7.5, "SSP_INI_TM_START Command". For example, after a LOGICAL_UNIT_RESET task management command (issued by the host), the host would need to send the SSP_ABORT command for all pending SSP I/Os associated with this logical unit. This is needed because the host, not the SPC 8x6G controller, has the knowledge of the logical unit. The host driver should not assume that the I/O request has been aborted until a response completion from the SPC 8x6G controller has been received.</p> <p>The SSP_ABORT command is always acknowledged by the SPC 8x6G through an SSP_ABORT Response described in Section 8.19.</p> <p>If the SPC 8x6G controller finds that there is valid SSP I/O associated with the HTAG-ABT, the SPC 8x6G will send an SSP_COMPLETION Response (Section 8.3) with STATUS set to IO_ABORTED, followed by an SSP_ABORT Response with STATUS set to IO_COMPLETED. If the SPC 8x6G controller finds that there is no valid SSP I/O associated with the HTAG-ABT, the SPC 8x6G will send an SSP_ABORT Response with STATUS set to IO_NOT_VALID.</p>
Aborting all SSP I/Os Associated with a Device ID	01b	<p>Typically this command is sent to abort all I/Os associated with DEVICE_ID following the host acknowledgement that the device has been removed. This abort all SSP I/Os for a device is typically done prior to removing the device handle through the DREGISTER_DEVICE_HANDLE Command described in Section 7.11.</p> <p>The SSP_ABORT Command is always acknowledged by the SPC 8x6G through an SSP_ABORT Response described in Section 8.19.</p> <p>If the SPC 8x6G controller finds that there are pending I/Os associated with DEVICE_ID, for each pending SSP I/O the SPC 8x6G will send an SSP_COMPLETION Response (Section 8.3) with STATUS set to IO_ABORTED, and finally followed by an SSP_ABORT Response with STATUS set to IO_COMPLETED. If the SPC 8x6G controller finds that there is no valid device associated with the DEVICE_ID, the controller will send an SSP_ABORT Response with STATUS set to IO_NOT_VALID.</p>

Usage

Initiator and target.

Command Format

Table 64 SSP_ABORT Command Format

	Byte 3				Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x00f							
1	HTAG														
2	DEVICE_ID														
3	HTAG of I/O to be aborted (HTAG-ABT)														
4	Reserved								NOQ	SCP					
5	Reserved														
...															
15															

Table 65 SSP_ABORT Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context of the abort operation. This tag is passed as the HTAG parameter in the SSP_ABORT Response described in Section 8.19.
2	Device Identifier	DEVICE_ID	4 Bytes	SSP device identifier. See Section 3.2, “ Device Handle and DEVICE_ID ” for a detailed description of DEVICE_ID.
3	HTAG of I/O to be aborted	HTAG-ABT	4 Bytes	Used only for aborting a single I/O. Tag or context for the I/O to be aborted. If the abort request is completed successfully, this tag is passed as the HTAG parameter in the SSP_COMPLETION with STATUS set to IO_ABORTED. See Section 8.3.
4 [2]	Resource Quarantine Requirement	NOQ	1 bit	Flag to indicate if I/O resource needs to be quarantined: 0: Requires I/O resource quarantine. 1: Do not quarantine the I/O resource.
4 [1:0]	Scope of Abort Flag	SCP	2 bits	Flag to indicate the scope of abort: 00b: Abort a single I/O described by the HTAG-ABT. 01b: Abort all I/Os associated with DEVICE_ID. The field HTAG-ABT is not used. Others: reserved.

7.11 DREGISTER_DEVICE_HANDLE Command

Description

A host sends this command to request the SPC 8x6G controller to remove all of the internal resources associated with the specified device handle previously registered through the [REGISTER_DEVICE Command](#) (Section 7.15) for initiator mode or indirectly registered through the [DEVICE_HANDLE_ACCEPT Command](#) (Section 7.7) for target mode.

All pending requests (SSP, SMP, and SATA) are completed (in the [SSP_COMPLETION Response](#) (Section 8.3), [SMP_COMPLETION Response](#) (Section 8.4), and [SATA_COMPLETION Response](#) (Section 8.9)) with STATUS set to IO_NO_DEVICE.

From the MPI perspective, the SPC 8x6G controller and the host exchange the device handle information using a DEVICE_ID field defined as index. For more information about device handle and DEVICE_ID, see Section 3.2, “[Device Handle and DEVICE_ID](#)”.

The completion response is reported in the [DREGISTER_DEVICE_HANDLE Response](#). See Section 8.7.

From the SCSI paradigm, a device handle represents the initiator/target nexus between an initiator and a target.

For a SAS target operation, the device handle represents the initiator/target nexus between a local target port and a remote initiator port. Since the SAS protocol does not have the concept of login and logout, the local target port may not detect the removal of the remote initiator port. Since the device handle is allocated from a limited resource pool, over time if the handles are not reclaimed from the stale nexus, the SPC 8x6G controller and the host eventually consumes all device handles from the pool and no I/O operation from a new initiator is done.

A different policy or scheme could be implemented to allow reclaiming the stale nexus. This specification assumes that the host would implement the device handle reclaiming policy. This function is used to allow the target host driver to inform the SPC 8x6G controller of a stale device handle and also to allow the host driver some level of control to which initiator it wants to respond.

For a SAS initiator operation the device handle represents the initiator/target nexus between the local initiator port and the remote target port. Since the device handle is allocated from a limited resource pool, the host initiator may want to manage which targets it needs to communicate with. This function is used to ask the SPC 8x6G controller to remove its internal resources for those target devices identified during discovery that the host does not intend to communicate with.

Usage

Initiator and target.

Command Format

Table 66 DREGISTER_DEVICE_HANDLE Command Format

	Byte 3			Byte 2		Byte 1		Byte 0			
0	V	H	R	BC=1			Reserved	OBID			
1	HTAG										
2	DEVICE_ID										
3	Reserved										
4	Reserved										
...											
15											

Table 67 DREGISTER_DEVICE_HANDLE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	Device to remove. See Section 3.2, “ Device Handle and DEVICE_ID ” for a detailed description of DEVICE_ID.
3	Reserved	Reserved	4 bits	Reserved.

7.12 GET_DEVICE_HANDLE Command

Description

This command is sent to get device handles (DEVICE_IDs) from a specific SAS/SATA local port in the SPC 8x6G controller. The completion of this command is reported in the [GET_DEVICE_HANDLE Response](#) described in Section 8.8.

Device handles are added as the result of device registration using the [REGISTER_DEVICE Command](#) described in Section 7.15.

A device handle is removed by sending a [DREGISTER_DEVICE_HANDLE Command](#) described in Section 7.11.

Usage

Initiator and target.

Command Format

Table 68 GET_DEVICE_HANDLE Command Format

	Byte 3			Byte 2		Byte 1		Byte 0				
0	V	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0011					
1	HTAG											
2	Reserved		DEVT	Maximum Device IDs (MDID)				Reserved PORT_ID				
3	SKIP COUNT (SC)											
4	Reserved											
...												
15												

Table 69 GET_DEVICE_HANDLE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [25:24]	Device Type	DEVT	2 bits	Type of device from which the DEVICE_ID is returned: Two-bit flag to specify the device type: 00b: STP device 01b: SSP or SMP device 10b: Directly-attached SATA device 11b: All device types
2 [23:8]	Max Device Identifiers	MAXID	2 Bytes	Maximum number of device handles (DEVICE_IDS) requested. Since the maximum size of the (concatenated) outbound IOMB for the response is limited to 1024 bytes, the maximum number requested device handles cannot exceed (256-3) entries.
2 [3:0]	Port Identifier	PORT_ID	4 bits	Identification for the SPC 8x6G's local SAS/SATA port.
3	SKIP COUNT	SC	4 bytes	This field is used to allow the host to request the GET_DEVICE_HANDLE Command in multiple sessions with each session specifying a different starting device handle list. A SKIP COUNT of zero indicates the beginning of the device handle maintained by the SPC 8x6G for the specified port. The DEVICE_IDC field in the GET_DEVICE_HANDLE Response (Section 8.8) indicates the number of device handles actually returned. The DEVICE_IDC field in the GET_DEVICE_HANDLE Response could be used as a reference for the SKIP COUNT field in the subsequent GET_DEVICE_HANDLE Command.

7.13 SMP_REQUEST Command

Description

This command sends an SMP request in initiator mode.

The completion response is reported in the [SMP_COMPLETION Response](#) described in Section [8.4](#).

Usage

Initiator.

Command Format

Table 70 SMP_REQUEST Command Format

	Byte 3			Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02	OPC=0x0012					
1	HTAG													
2	DEVICE_ID													
3	Reserved			LEN		Reserved			IP IR					
4	SMP_REQUEST Frame (SMPRF[15:0])													
5														
6														
7														
8	Indirect SMP Payload Address Low (ISPAL) or SMPRF[19:16]													
9	Indirect SMP Payload Address High (ISPAH) or SMPRF[23:20]													
10	Indirect SMP Payload Length (ISPL) or SMPRF[27:24]													
11	Reserved or SMPRF[31:28]													
12	Indirect SMP_RESPONSE Frame Address Low (ISRAL) or SMPRF[35:32]													
13	Indirect SMP_RESPONSE Frame Address High (ISRAH) or SMPRF[39:36]													
14	Indirect SMP_RESPONSE Frame Length (ISRL) or SMPRF[43:40]													
15	RSP Reserved or SMPRF[47:44]													

Table 71 SMP_REQUEST Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	Device Identifier	DEVICE_ID	4 Bytes	SMP target device identifier. See Section 3.2 for a detailed description of DEVICE_ID.
3 [23:16]	Length	LEN	1 Byte	Number of bytes for the direct SMP request frame. Valid only if IP (Indirect Payload) is set to 0b. The maximum SMP request length that can be sent in direct mode is 48 bytes.
3 [1]	Indirect Payload	IP	1 bit	Indicates the location of the SMP_REQUEST frame to send: 0b: Use the inbound IOMB for the SMP_REQUEST frame. Only for SMP_REQUEST frames up to 48 bytes long. DWord [7:4] contain the first 16 bytes of the SMP_REQUEST frame. 1b: Use DWord [11:8] as the physical address in host memory where the SMP_REQUEST frame to send resides.
3 [0]	Indirect Response	IR	1 bit	Indicates the location for the SMP_RESPONSE frame to be received: 0b: Use the outbound IOMB for the SMP_RESPONSE frame. DWWords [15:2] are not used. 1b: Use DWWords [15:12] as the physical address in host memory where the SPC 8x6G writes the received SMP_RESPONSE frame.
4	SMP_REQUEST Frame	SMPRF	16 Bytes	The first 16 bytes of the SMP_REQUEST frame to send. Valid only if the IP flag is set to 0b. Big Endian format. See Section 2.1.3.1, "SAS Payload Endianness" for details about SAS payload endianness.
8	Indirect SMP_REQUEST Frame address Low	ISPAL	4 Bytes	If the IP flag is set to 1b, this field contains the lower 32-bit physical address, in host memory, of the buffer that contains the SMP_REQUEST frame. If the IP flag is set to 0b this field contains the SMP_REQUEST frame bytes 16 to 19).
9	Indirect SMP_REQUEST Frame address High	ISPAH	4 Bytes	If the IP flag is set to 1b, this field contains the higher 32-bit physical address, in host memory, of the buffer that contains the SMP_REQUEST frame. If the IP flag is set to 0b this field contains the SMP_REQUEST frame bytes 20 to 23).
10	Indirect SMP_REQUEST Frame Length	ISPL	4 Bytes	If the IP flag is set to 1b, this field contains the size of the SMP_REQUEST frame. If the IP flag is set to 0b this field contains the SMP_REQUEST frame bytes 24 to 27.
11	Reserved		4 Bytes	If the IP flag is set to 1b, this field is reserved. If the IP flag is set to 0b this field contains the SMP_REQUEST frame bytes 28 to 31.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
12	Indirect SMP_RESPONSE Frame Address Low	ISRAL	4 Bytes	If the IR flag is set to 1b, this field contains the lower 32-bit physical address, in host memory, of the buffer that contains the SMP_RESPONSE frame. If both the IR flag and IP flag are set to 0b, this field contains the SMP_REQUEST frame bytes 32 to 35.
13	Indirect SMP_RESPONSE Frame Address High	ISRAH	4 Bytes	If the IR flag is set to 1b, this field contains the higher 32-bit physical address, in host memory, of the buffer that contains the SMP_RESPONSE frame. If both the IR flag and IP flag are set to 0b, this field contains the SMP_REQUEST frame bytes 35 to 39.
14	Indirect SMP_RESPONSE Frame Address Length	ISRL	4 Bytes	If the IR flag is set to 1b, this field contains the size of the SMP_RESPONSE frame. If both the IR flag and IP flag are set to 0b, this field contains the SMP_REQUEST frame bytes 40 to 43.
15	Reserved	RSV	4 Bytes	If either the IR flag or IP flag are set to 1b, this field is reserved. If both the IR flag and IP flag are set to 0b, this field contains the SMP_REQUEST frame bytes 44 to 47.

7.14 SMP_ABORT Command

Description

This command is sent to abort an SMP command/response operation due to a port invalid condition (all PHYs are down) that was previously initiated by the following command:

- [SMP_REQUEST Command](#) (Section 7.13)

The affected SMP operation is aborted and reported in the [SMP_COMPLETION Response](#) (Section 8.4) with STATUS set to IO_ABORTED.

The SPC 8x6G supports a hardware-based SMP timer that will automatically report the SMP error with the status set to IO_ERROR_HW_TIMEOUT in the [SMP_COMPLETION Response](#) when a timeout occurs. The [SMP_ABORT Command](#) cannot be used to abort an SMP operation that has already been sent to the wire. The [SMP_ABORT Command](#) should only be used to clean-up SPC 8x6G internal resources following a PHY down event that results in an invalid port.

The completion of this command is reported in the [SMP_ABORT Response](#) as described in Section 8.27.

The SMP_ABORT command has an option field (SCP) to indicate the scope of abort: to abort an individual SMP operation with a specific HTAG-ABT or to abort all SMP operations associated with a DEVICE_ID.

Table 72 SMP_ABORT Scope of Abort (SCP)

Option	SCP Encoding	Description
Aborting a Single SMP operation	00b	<p>Typically this command is sent to abort an SMP operation due to a port invalid condition (all PHYs are down).</p> <p>This abort operation is typically done prior to removing the device handle through the DEREGISTER_DEVICE_HANDLE Command described in Section 7.11.</p> <p>The SMP_ABORT command is always acknowledged by the SPC 8x6G through an SMP_ABORT Response (Section 8.27).</p> <p>If the SPC 8x6G controller finds that there is valid SMP operation associated with the HTAG-ABT, the SPC 8x6G will send an SMP_COMPLETION Response (Section 8.4) with STATUS set to IO_ABORTED, followed by an SMP_ABORT Response with STATUS set to IO_COMPLETED. If the SPC 8x6G controller finds that there is no valid SMP operation associated with the HTAG-ABT, the SPC 8x6G will send an SMP_ABORT Response with STATUS set to IO_NOT_VALID.</p>

Option	SCP Encoding	Description
Aborting all SMP operations Associated with a Device ID	01b	<p>Typically this command is sent to abort all SMP operations associated with DEVICE_ID due to a port invalid condition (all PHYs are down).</p> <p>This abort operation is typically done prior to removing the device handle through the DEREGISTER_DEVICE_HANDLE Command.</p> <p>The SMP_ABORT command is always acknowledged by the SPC 8x6G through an SMP_ABORT Response.</p> <p>If the SPC 8x6G controller finds that there are pending SMP operations associated with DEVICE_ID, for each pending SMP operation the SPC 8x6G will send an SMP_COMPLETION Response with STATUS set to IO_ABORTED, and finally followed by an SMP_ABORT Response with STATUS set to IO_COMPLETED. If the SPC 8x6G controller finds that there is no valid device associated with the DEVICE_ID, the controller will send an SMP_ABORT Response with STATUS set to IO_NOT_VALID.</p>

Usage

Initiator and target.

Command Format

Table 73 SMP_ABORT Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0014
1	HTAG							
2	DEVICE_ID							
3	HTAG of I/O to be aborted (HTAG-ABT)							
4	Reserved							SCP
5								
...								
15	Reserved							

Table 74 SMP_ABORT Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for the SMP operation to be aborted. This field is the TAG value previously passed in a SMP_REQUEST Command .

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	Device Identifier	DEVICE_ID	4 Bytes	SAS device identifier. See Section 3.2, "Device Handle and DEVICE_ID" for a detailed description of DEVICE_ID.
3	HTAG of I/O to be aborted	HTAG-ABT	4 Bytes	Used only for aborting a single SMP operation. Tag or context for the SMP operation to be aborted. If the abort request is completed successfully, this tag is passed as the HTAG parameter in the SMP_COMPLETION Response with STATUS set to IO_ABORTED.
4 [1:0]	Scope of Abort Flag	SCP	2 bits	Flag to indicate the scope of abort: 00b: Abort a single SMP operation described by the HTAG-ABT. 01b: Abort all SMP operation associated with DEVICE_ID. The field HTAG-ABT is not used. Others: reserved.

7.15 REGISTER_DEVICE Command

Description

This command is only sent to register a new device that the host has discovered on its own or the directly-attached SAS/SATA device that is reported during PHY up in a [SAS_HW_EVENT Notification](#), which is described in Section 8.2. The host implements its own SAS discovery scheme (that is, by sending a direct SMP request).

The completion of the device registration is reported in the [DEVICE_REGISTRATION Response](#), where the SPC 8x6G-assigned DEVICE_ID is returned for the corresponding device. See Section 8.6.

The DEVICE_ID is removed using the [DEREGISTER_DEVICE_HANDLE Command](#). See Section 7.11.

A device handle registered using a particular PORT_ID should not be reused if the device is physically moved to a different port (PORT_ID), without deregistering and re-registering it again with a correct PORT_ID.

Note: Registering the same device multiple times using different PORT_ID fields is allowed. However, registering the same device multiple times using the same PORT_ID field is not allowed.

Usage

Initiator.

Command Format

Table 75 REGISTER_DEVICE Command Format

	Byte 3				Byte 2		Byte 1		Byte 0												
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0016													
1	HTAG																				
2	Reserved									PHYID	PORT_ID										
3	Reser ved	S	DLR	Reserved		Reserved				A	HA	R									
4	Reserved					IT Nexus Timeout (ITNT)															
5	SAS Address High (SADDRH)																				
6	SAS Address Low (SADDRL)																				
7	UPPER DEVICE_ID					Reserved															
8	Reserved																				
...																					
15																					

Table 76 REGISTER_DEVICE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [7:4]	PHY Identifier	PHYID	4 bits	This field is used only if S flag is 10b, directly-attached SATA drive. Zero-based PHY identifier.
2 [3:0]	Port Identifier	PORT_ID	4 bits	Identification for the SPC 8x6G local SAS/SATA port.
3 [29:28]	SAS/SATA Device Type	S	2 bits	Two bit S flag to specify device type: 00b: STP device 01b: SSP or SMP device 10b: Directly-attached SATA device
3 [27:24]	Device Link Rate	DLR	4 bits	The controller uses this value (of the Open Address frame) for the Connection Rate field when opening the device: 0x08: 1.5 Gbit/s 0x09: 3.0 Gbit/s 0x0A: 6.0 Gbit/s All others reserved.
3 [15:2]	Reserved	Reserved	14 bits	Reserved.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
3 [2]	AWT flag	A	1 bit	<p>Priority setting for the Arbitration Wait Time (AWT) for this remote device:</p> <p>0b: Default setting (recommended). The actual AWT value is based on how long an OPEN frame has been waiting for a connection request to be accepted. It starts at 0. As specified in the SAS specification, from 0 to 32768 µs, the AWT value is incremented every µs. From 32768 µs and on, the AWT value is incremented every ms.</p> <p>1b: Increase priority. The actual AWT value starts at 32768 µs when the A flag is set to '1'. The AWT value is incremented every ms from that point on.</p>
3 [1]	Host Assigned Upper DEVICE_ID	HA	1 bit	<p>When set, indicates that the host will assign the upper 16 bits of the DEVICE_ID in the UPPER DEVICE_ID field. The 32-bit DEVICE_ID reported in the DEVICE_REGISTRATION Response (Section 8.6) will contain the value specified in the UPPER DEVICE_ID field in the upper 16 bits and the SPC 8x6G-assigned value in the lower 16 bits.</p> <p>When cleared, the SPC 8x6G will assign the full 32 bits of the DEVICE_ID that will be reported in the DEVICE_REGISTRATION Response.</p>
3 [0]	Retry flag	R	1 bit	<p>This field enables the Transport Layer Retry (TLR) flag per SAS 1.1 and SAS 2.0:</p> <p>1b: Enable TLR 0b: Disable TLR PMC-Sierra recommends that TLR is enabled (set to 1b).</p> <p>See Section 3.20, "Transport Layer Retry (TLR) Handling".</p>
4 [15:0]	IT Nexus Timeout	ITNT	2 Bytes	The value in milliseconds of the time unit that is used by the SPC 8x6G to determine the nexus timeout condition.
5	SAS Address High	SADDRH	4 Bytes	<p>This field is used only if the S flag is 00b or 01b.</p> <p>High 32 bits of the Device SAS address in Big Endian format.</p>
6	SAS Address Low	SADDRL	4 Bytes	<p>This field is used only if the S flag is 00b or 01b.</p> <p>Low 32 bits of the Device SAS address in Big Endian format.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
7	Upper 16-bit of DEVICE_ID	UPPER DEVICE_ID	2 Bytes	<p>This field is only used when the HA field is set.</p> <p>This is the host-assigned in the upper 16 bits in the DEVICE_ID.</p> <p>The host can assign bits [15:0] for its own purposes.</p> <p>The 32-bit DEVICE_ID reported in the DEVICE_REGISTRATION Response will contain the value specified in the UPPER DEVICE_ID field in the upper 16 bits and the SPC 8x6G-assigned value in the lower 16 bits.</p> <p>If the HA bit is not set, this field is reserved and should be set to zero.</p>

7.16 SATA_HOST_IO_START Command

Description

The SATA Host I/O Operation Start command is sent to initiate a SATA or STP request encapsulated inside the Register Host-to-Device FIS type.

The SPC 8x6G controller supports PIO, legacy DMA, FPDMA, and non-data FIS.

The host is responsible for the I/O serialization if I/Os are mixed between FPDMA (NCQ) and other types to the same SATA drive.

The completion to this command is reported in the [SATA_COMPLETION Response](#) described in Section 8.9. See Section 3.13, “[SATA Host Write Operations](#)” and Section 3.14, “[SATA Host Read Operations](#)” for details.

Usage

Initiator.

Command Format

Table 77 SATA_HOST_IO_START Command Format

	Byte 3				Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0017							
1	HTAG														
2	DEVICE_ID														
3	DL														
4	Reserved		RE	TFI	S	NCQTAG	ATAP	DIR	Reserved	M Reser ved					
5	SATA Register Host to Device FIS (SATAFIS)														
...															
9															
10	Reserved														
11															
12	SGLAL														
13	SGLAH														
14	LEN														
15	E	Reserved													

Table 78 SATA_HOST_IO_START Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	SATA or STP target device identifier. See Section 3.2 for a detailed description of DEVICE_ID.
3	Data Length	DL	4 Bytes	Expected data transfer length.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [24]	Return FIS on Good Completion	RETFIS	1 Bit	<p>This bit provides an option for the host to ask the SPC 8x6G to return the FIS reported by the device even when the device does not report an error. Under normal “no error” conditions, no FIS will be returned to the host unless this bit is set.</p> <p>0b: No FIS will be return 1b: FIS will be returned</p> <p>The type of FIS returned depends on the type of SATA protocol used (PIO, DMA, FPDMA or non-data):</p> <p>PIO Read: Nothing is returned PIO Write: Device To Host FIS received from the device. DMA Read: Device To Host FIS received from the device. DMA Write: Device To Host FIS received from the device. FPDMA Read: Set Device Bit FIS received from the device. FPDMA Write: Set Device Bit FIS received from the device. Non-Data: Device To Host FIS received from the device.</p>
4 [23:16]	SATA Native Command Queue Tag	NCQTAG	1 Byte	Only uses a SATA Native Command Queue tag when the ATA PROT is set NCQ (FPDMA). This NCQTAG field supersedes the NCQ TAG field in the DWord [9:5] SATA FIS fields.
4 [15:10]	ATA Protocol	ATAP	6 bits	The ATA protocol type: 0x01: SATA_PROTOCOL_SRST_ASSERT 0x02: SATA_PROTOCOL_SRST_DEASSERT 0x03: SATA_PROTOCOL_EXECDEVDIAG 0x04: SATA_PROTOCOL_NONDATA 0x05: SATA_PROTOCOL_PIO 0x06: SATA_PROTOCOL_DMA 0x07: SATA_PROTOCOL_FPDMA
4 [9:8]	Direction	DIR	2 bits	Direction of data transfer: 00b: No data transfer 01b: Data transfer from target to initiator 10b: Data transfer from initiator to target

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [2]	Message Report	M	1 bit	<p>This bit provides the option for the host to ask the SPC 8x6G to send an event to the host when the frame has been sent on the wire.</p> <p>The event IO_XFER_CMD_FRAME_ISSUED in the SATA_EVENT Notification is used to notify the host that the frame has been sent and an R_OK primitive has been received. See Section 8.10.</p> <p>1b: Send the event after the frame is sent 0b: Do not send event after the frame is sent. This is the default setting.</p>
9:5	SATA Register Host to Device FIS	SATAFIS	20 Bytes	SATA Register Host to Device FIS in Little Endian format. See 2.1.3.2 for details about SATA payload endianness.
12	SGL Address Low	SGLAL	4 Bytes	<p>If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory.</p> <p>If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory.</p>
13	SGL Address High	SGLAH	4 Bytes	<p>If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory.</p> <p>If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory.</p>
14	Length	LEN	4 Bytes	<p>If Local SGL is used, this field contains the size in bytes of the data buffer in host memory.</p> <p>If ESGL is used, this field is not used.</p>
15 [31]	Extension bit	E	1 bit	<p>This bit indicates if this SGL element (L/A pair) describes a data buffer or an ESGL in host memory:</p> <p>0b: This SGL element describes a data buffer 1b: This SGL element describes an ESGL</p>

7.17 SATA_ABORT Command

Description

This command is sent to abort one or more I/O requests previously initiated by the [SATA_HOST_IO_START Command](#) described in Section [7.16](#).

This command only aborts or invalidates pending I/Os in the SPC 8x6G controller and does not send any protocol, on the link, to abort an I/O on the other side.

The SATA_ABORT command has an option field (SCP) to indicate the scope of abort: to abort an individual I/O with a specific HTAG-ABT or to abort all I/Os associated with a DEVICE_ID.

If NOQ is not set, the SPC 8x6G controller releases I/O resources for the aborted SATA I/O after a delay which is two times the IT Nexus timeout for the device.

Table 79 SATA_ABORT Scope of Abort (SCP)

Option	SCP Encoding	Description
Aborting a Single SATA I/O	00b	<p>For SATA, typically using the abort all I/Os for a DEVICE_ID is a more convenient way. For flexibility purposes, the abort for a single I/O option is provided.</p> <p>The SATA_ABORT Command is always acknowledged by the SPC 8x6G through a SATA_ABORT Response. See Section 8.20.</p> <p>If the SPC 8x6G controller finds that there is valid I/O associated with the HTAG-ABT, it will send a SATA_COMPLETION Response (Section 8.9) with STATUS set to IO_ABORTED, followed by a SATA_ABORT Response with STATUS set to IO_COMPLETED.</p> <p>If the SPC 8x6G controller finds that there is no valid I/O associated with the HTAG-ABT, it will send SATA_ABORT Response with STATUS set to IO_NOT_VALID</p>
Aborting all SATA I/Os Associated with DEVICE_ID	01b	<p>Typically this command is sent:</p> <ul style="list-style-type: none"> After the completion of a successful link reset sequence or a SATA Soft Reset to abort the corresponding I/Os in SATA device, or Following the SATA_EVENT for a SATA NCQ error with the event set to IO_XFER_ERROR_ABORTED_NCQ_MODE. Prior to removing a device handle using the DEREGISTER_DEVICE_HANDLE Command described in Section 7.11, to abort all I/Os associated with the device handle. <p>The host driver should not assume that the I/O request has been aborted until a response completion from the SPC 8x6G controller has been received.</p> <p>The SATA_ABORT Command is always acknowledged by the SPC 8x6G through a SATA_ABORT Response.</p> <p>If the SPC 8x6G controller finds that there are pending I/Os associated with DEVICE_ID, for each pending I/O it will send a</p>

Option	SCP Encoding	Description
		<p>SATA_COMPLETION Response with STATUS set to IO_ABORTED, finally followed by a SATA_ABORT Response with STATUS set to IO_COMPLETED.</p> <p>If the SPC 8x6G controller finds that there is no valid device associated with the DEVICE_ID, it will send a SATA_ABORT Response with STATUS set to IO_NOT_VALID.</p>

Usage

Initiator.

Command Format

Table 80 SATA_ABORT Command Format

	Byte 3				Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0018							
1	HTAG														
2	DEVICE_ID														
3	HTAG of I/O to be aborted (HTAG-ABT)														
4	Reserved							NOQ	SCP						
5	Reserved														
...															
15															

Table 81 SATA_ABORT Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context of the abort operation. This tag is passed as the HTAG parameter in the SATA_ABORT Response .
2	Device Identifier	DEVICE_ID	4 Bytes	SATA or STP target device identifier. See Section 3.2, “ Device Handle and DEVICE_ID ” for a detailed description of DEVICE_ID.
3	HTAG of I/O to be aborted	HTAG-ABT	4 Bytes	Used only for aborting a single I/O. Tag or context for the I/O to be aborted. If the abort request is completed successfully, this tag is passed as HTAG parameter in SATA_COMPLETION with STATUS set to IO_ABORTED. See Section 8.9.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [2]	Resource Quarantine Requirement	NOQ	1 bit	Flag to indicate if I/O resource needs to be quarantined: 0: Requires I/O resource quarantine. 1: Do not quarantine the I/O resource.
4 [1:0]	Scope of Abort Flag	SCP	2 bits	Flag to indicate the scope of abort: 00b: Abort a single I/O described by the HTAG-ABT. 01b: Abort all I/Os associated with DEVICE_ID. The field HTAG-ABT is not used. Others: Reserved.

7.18 LOCAL_PHY_CONTROL Command

Description

This command is sent to initiate a PHY_CONTROL command to the SPC 8x6G controller's local PHY. The completion of this command is reported in the [LOCAL_PHY_CONTROL Response](#) described in Section 8.5.

Usage

Initiator and target.

Command Format

Table 82 LOCAL_PHY_CONTROL Command Format

	Byte 3				Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved		OBID	CAT=0x02	OPC=0x0019						
1	HTAG														
2	Reserved				PHYOP			Reserved	PHYID						
3	Reserved														
...															
15															

Table 83 LOCAL_PHY_CONTROL Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [3:0]	PHY Identifier	PHYID	4 bits	Zero based PHY identifier.
2 [15:8]	PHY Operation	PHYOP	1 Byte	<p>PHY operation:</p> <p>0x01: LINK RESET. Perform a link-reset sequence on the specified PHY and enable the specified PHY. The PHY bypasses the SATA spin-up hold state. This operation causes the IOP_EVENT_PHY_DOWN event reported in the SAS_HW_EVENT Notification. See Section 8.2.</p> <p>0x02: HARD RESET. Perform a link-reset sequence on the specified PHY and enable the specified PHY. If the attached PHY is a SAS PHY or an expander PHY, the link reset sequence includes a hard reset sequence. If the attached PHY is a SATA PHY, the PHY bypasses the SATA spin-up hold state. This operation causes the IOP_EVENT_PHY_DOWN event reported in the SAS_HW_EVENT Notification.</p> <p>0x10: TX NOTIFY SPINUP: Send NOTIFY (ENABLE SPINUP) primitive to clear the spin up hold on the PHY.</p> <p>0x12: TX BROADCAST ASYNCHRONOUS EVENT: Send a BROADCAST (ASYNCHRONOUS EVENT) primitive on the PHY.</p> <p>0x20: COMINIT_OOB: Send a COMINIT OOB on the PHY. This command is used for vendor-specific OOB protocols used to communicate to external devices such as expanders. Before issuing this command, the PHY must be disabled by sending a PHY_STOP Command for the corresponding PHY.</p>

7.19 GET_DEVICE_INFO Command

Description

This command is sent to get the device detail information associated with a DEVICE_ID.

The completion of this command is reported in the [GET_DEVICE_INFO Response](#) described in Section [8.14](#).

Usage

Initiator and target.

Command Format

Table 84 GET_DEVICE_INFO Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x001a
1	HTAG							
2	DEVICE_ID							
3	Reserved							
...								
15								

Table 85 GET_DEVICE_INFO Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	SATA or SSP device identifier. See Section 3.2, “Device Handle and DEVICE_ID” for a detailed description of DEVICE_ID.

7.20 FW_FLASH_UPDATE Command

Description

This command is sent to initiate an SPC 8x6G firmware flash image and customer-specific Expansion ROM flash image to update to the SPC 8x6G controller. This operation can only be performed if operational SPC 8x6G firmware is used on an SPC 8x6G device that has flash memory.

The operational SPC 8x6G firmware will detect using the image header which flash partition the image is to be written to: the flash partition for the SPC 8x6G firmware or the flash partition for the Expansion ROM.

Due to the relatively large flash image size and host memory requirements, the host performs the firmware flash update in multiple pieces. Note that subsequent flash update pieces should only be initiated after the prior piece of the flash update is completed. Note each update piece uses a single SGL up to 4 Kbytes in size.

The Expansion ROM code is not provided by PMC-Sierra and is the responsibility of the customer. However, in order for the Expansion ROM image to be written to the flash memory, a PMC-Sierra specific utility is required to format the image. Please refer to the SPC-8x6G firmware release notes document for details.

The completion of each of this operation is reported through a [FW_FLASH_UPDATE Response](#) described in Section 8.15.

Usage

Initiator and target.

IOMB Format

Table 86 FW_FLASH_UPDATE Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0020
1	HTAG							
2	CURRENT IMAGE OFFSET (C_OFFSET)							
3	CURRENT IMAGE LENGTH (C_LEN)							
4	TOTAL IMAGE LENGTH (T_LEN)							
5	Reserved							
...								
11								

	Byte 3	Byte 2	Byte 1	Byte 0
12		SGLAL		
13		SGLAH		
14		LEN		
15	E	Reserved		

Table 87 FW_FLASH_UPDATE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Current Image Offset	C_OFFSET	4 Bytes	The image offset of the current operation. Flash image blocks are transferred from the host to the SPC 8x6G sequentially without any gaps.
3	Current Image Length	C_LEN	4 Bytes	The length in bytes of the current image blocks transfer. This field shall be $\leq 4K$. The minimum size is 28 bytes, which is the length of the flash memory header.
4	Total Image Length	T_LEN	4 Bytes	Total flash image length. The host sets this value at the initial flash image update operation when C_OFFSET is 0. The minimum size is 28 bytes, which is the length of the flash memory header.
12	SGL Address Low	SGLAL	4 Bytes	If Local SGL is used, this field contains the lower 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the lower 32 physical address bits of the SGL in host memory.
13	SGL Address High	SGLAH	4 Bytes	If Local SGL is used, this field contains the higher 32 physical address bits for the data buffer in host memory. If ESGL is used, this field contains the higher 32 physical address bits of the SGL in host memory.
14	Length	LEN	4 Bytes	If Local SGL is used, this field contains the size in bytes of the data buffer in host memory. If ESGL is used, this field is not used. Currently, only Local SGL up to 4 Kbytes is supported.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
15 [31]	Extension bit	E	1 bit	This bit indicates if this SGL element (L/A pair) describes a data buffer or an ESGL in host memory: 0b: This SGL element describes a data buffer 1b: This SGL element describes an ESGL Currently, only Local SGL is supported.

7.21 GPIO Command

Description

This command manages the General Purpose Input/Output (GPIO) signals available in the SPC 8x6G device (See Section 3.16, “[GPIO Operation](#)”.)

The completion for this command is reported in the [GPIO Response](#) as described in Section 8.16.

Usage

Initiator and target.

Command Format

Table 88 GPIO Command Format

	Byte 3			Byte 2		Byte 1		Byte 0							
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x022							
1	HTAG														
2	Reserved			EOBID		Reserved			GE GS GR G W						
3	Reserved			GPIOWRMSK											
4	Reserved			GPIOWRVAL											
5	Reserved			GPIOIE											
6	Reserved		OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0	
7	Reserved			Reserved	OT19	OT18	OT17	OT16	OT15	OT14	OT13	OT12			
8	Reserved			GPIEVCHANGE											
9	Reserved			GPIEVRISE											
10	Reserved			GPIEVFALL											
11	Reserved														
...															
15															

Table 89 GPIO Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2 [20:16]	Event OBID	EOBID	5 bits	Outbound Queue used to send the GPIO event.
2 [3]	GPIO Event Setup	GE	1 bit	0b: No GPIO event setup is done. The GPIEVCHANGE, GPIEVRISE, GPIEVFALL, and EOBID fields are not valid. 1b: GPIO event setup is done. The GPIEVCHANGE, GPIEVRISE, GPIEVFALL, and EOBID fields are valid.
2 [2]	GPIO Pin Setup	GS	1 bit	0b: No GPIO pin set-up operation is done. GPIOIE and GPIOTYPE are not valid fields. 1b: A GPIO setup operation is done. GPIOIE and GPIOTYPE are valid fields.
2 [1]	GPIO Read	GR	1 bit	GPIO Read operation. 0b: No GPIO read operation is done in this command. 1b: This command enables a GPIO read operation. The GPIORDVAL fields in the GPIO Response contain the read value from the GPIO ports. The GPIOIE, GPIEVCHANGE, GPIEVRISE, and GPIEVFALL fields in the GPIO Response return the values set up for the GPIO system.
2 [0]	GPIO Write	GW	1 bit	GPIO Write operation. 0b: No GPIO write operation is done in this command. Fields GPIOWRMSK and GPIOWRVAL are not valid. 1b: This command does a GPIO write operation. Fields GPIOWRMSK and GPIOWRVAL are valid. If the GPIO input signal is not enabled, the value written will not be saved by the SPC 8x6G. That is, the next GPIO read will not reflect the value written. To read the value written to a output pin, the corresponding input bit has to be enabled for input.
3 [19:0]	GPIO Write Mask	GPIOWRMSK	20 bits	When the GW field is 1b, this field contains the mask that is applied to the write operation. Each bit set to 1b in the mask lets the corresponding bit value in the GPIOWRVAL be transferred to the GPIO output signal.
4 [19:0]	GPIO Write Value	GPIOWRVAL	20 bits	When the GW field is 1b, this field contains the bit value that is transferred to the GPIO output signal when the corresponding bit in the GPIOWRMSK field is set to 1b.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
5 [19:0]	GPIO Input Enabled	GPIOIE	20 bits	When the GS field is set to 1b, this field is a bitmap that configures the direction of the corresponding GPIO pin. 0b: The corresponding GPIO pin is not input enabled. Output is always enabled. 1b: The corresponding GPIO pin is input enabled. Output is always enabled.
6 [23:0]	GPIO Type	OT[0..11]	12x2 bits	When the GS field is set to 1b, for GPIO signals 0 to 11, this field configures how the output enable is driven. 00b: Tristated (default) 01b: Driven 10b: Reserved 11b: Reserved
7 [15:0]	GPIO Type	OT[12..19]	8x2 bits	When the GS field is set to 1b, for GPIO signals 12 to 19, this field configures how the output enable is driven. 00b: Tristated 01b: Driven 10b: Reserved 11b: Reserved Signals 12 to 19 are reserved by the SPC 8x6G. (See Section 3.16, “GPIO Operation”.)
8 [19:0]	GPIO Event Level	GPIOEVCHAN GE	20 bits	When the GE bit is set to 1b, this field is a bitmap that enables a GPIO event: 0b: A level change does not trigger a GPIO event 1b: A level change triggers a GPIO event
9 [19:0]	GPIO Event Rising Edge	GPIOEVRISE	20 bits	When the GE bit is set to 1b, this field is a bitmap that enables a GPIO event: 0b: Rising edge does not trigger a GPIO event 1b: Rising edge triggers a GPIO event
10 [19:0]	GPIO Event Falling Edge	GPIOEVFALL	20 bits	When the GE bit is set to 1b, this field is a bitmap that enables a GPIO event: 0b: Falling edge does not trigger a GPIO event 1b: Falling edge triggers a GPIO event

7.22 SAS_DIAG_MODE_START_END Command

Description

This command is sent to the SPC 8x6G controller to either start or end the SAS diagnostic mode of one of the PHYs in the SPC 8x6G device. The completion of each start/end operation is reported through a [SAS_DIAG_MODE_START_END Response](#) as described in Section 8.21.

Prior to entering SAS diagnostic mode, if the PHY has been previously started using the [PHY_START Command](#) described in Section 7.2, depending on the SAS diagnostic operation, the host is required to stop the PHY from normal operation by issuing the [PHY_STOP Command](#) described in Section 7.3. After all of the responses are received with an indication that the PHY has been stopped, the host can issue this command to start diagnostic mode.

When the SPC 8x6G receives SAS_DIAG_MODE_START_END (START) Command, it will start the PHY in diagnostic mode. The supported link rate is the one set in previous [PHY_START Command](#).

Please see the CMD_TYPE description in the [SAS_DIAG_EXECUTE Command](#) description (Section 7.23) to determine which command type requires the PHY to be put into a SAS diagnostic mode first.

This command is also issued to end the SAS diagnostic mode of an SPC 8x6G PHY. A [PHY_START Command](#) should only be issued to restart normal operation after a [SAS_DIAG_MODE_START_END Response](#) is received indicating the termination of the SAS diagnostic phase of a particular PHY.

Usage

Initiator and target.

IOMB Format

Table 90 SAS_DIAG_MODE_START_END Command Format

	Byte 3				Byte 2		Byte 1		Byte 0								
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0023									
1	HTAG																
2	Reserved				OP		Reserved	PHYID									
3	Reserved																
...																	
15																	

Table 91 SAS_DIAG_MODE_START_END Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [15:8]	Start or end Operation	OP	1 Byte	0x00: end SAS diagnostic mode 0x01: start SAS diagnostic mode
2 [3:0]	PHY Identifier	PHYID	4 bits	Zero-based PHY Identifier.

7.23 SAS_DIAG_EXECUTE Command

Description

This command is sent to the SPC 8x6G to initiate one of the SAS diagnostic tasks. The completion of this operation is reported using a [SAS_DIAG_EXECUTE Response](#) as described in Section 8.22.

Prior to executing a SAS diagnostic command, the host is required to enter the particular SPC 8x6G PHY into SAS diagnostic mode by sending a [SAS_DIAG_MODE_START_END Command](#) with an option to enter diagnostic mode. See Section 7.22.

After completing the SAS diagnostic command(s), if the host desires to bring the PHY back to normal operation mode, the host is required to terminate the particular SPC 8x6G PHY's SAS diagnostic mode by sending a [SAS_DIAG_MODE_START_END Command](#) with an option to end the diagnostic mode.

A SAS diagnostic action is specified by a command, which is a combination of a command type (CMD_TYPE) and a command descriptor (CMD_DESC).

[Table 94](#) lists the valid combinations of command types and command descriptors. The numeric assignment of the command descriptor is command type specific. [Table 95](#) and [Table 96](#) list the command descriptor numeric assignment based on command types.

Some commands can be run only in diagnostic mode:

- CMD_TYPE = DIAG_OPRN_PERFORM: all commands except for CODE_VIOL_INSERT and DISP_ERR_INSERT.
- CMD_TYPE = DIAG_OPRN_STOP
- CMD_TYPE = RECEIVE_ENABLE

Some of the diagnostic commands also include additional parameters in IOMB DWords [6:3].
[Table 97](#) below lists the SAS diagnostic commands that require an argument.

Usage

Initiator and target.

IOMB Format

Table 92 SAS_DIAG_EXECUTE Command Format

	Byte 3			Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02	OPC=0x0024					
1	HTAG													
2	Reserved			CMD_TY PE		CMD_DESC		Reserved	PHYID					
3	Reserved		PAT1		Reserved		PAT2							
4	Reserved						THRSHLD							
5	Reserved		CODE_VIOL_PAT		Reserved		PRBS_ERR_MSK							
6	PMON													
7	PERF1CTL													
8	Reserved													
...														
15														

Table 93 SAS_DIAG_EXECUTE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [15:13]	Command Type	CMD_TYPE	3 bits	000b: DIAG_OPRN_PERFORM: Perform an operation e.g. continuously insert generated test pattern, configure a loop back, insert an error, etc. This command type requires the PHY to be put into SAS diagnostic mode first. 001b: DIAG_OPRN_STOP: Stop/unconfigure e.g. stop inserting generated test pattern, unconfigure a loop back etc. This command type requires the PHY to be put into SAS diagnostic mode first. 010b: THRESHOLD_SPECIFY: Specify a threshold for an error count e.g. PRBS error count, disparity error count, code violation error count. This command type requires the PHY to be put into SAS diagnostic mode first.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>011b: RECEIVE_ENABLE: Enable receiving of a certain kind of pattern on a PHY so that the received patterns are accounted. Required for PRBS and CJTPAT. This command type requires the PHY to be put into SAS diagnostic mode first.</p> <p>100b: DIAG_REPORT_GET: Get a report e.g. PRBS error count, lost DWord sync count, etc. This command type does not require the PHY to be put into SAS diagnostic mode first.</p> <p>101b: ERR_CNT_RESET: Reset an error counter so that it counts afresh. This command type does not require the PHY to be put into SAS diagnostic mode first.</p>
2 [12:8]	Command Descriptor	CMD_DESC	5 bits	See Table 94 , Table 95 , and Table 96 , which list the command descriptor numeric assignment based on command types.
2 [3:0]	PHY Identifier	PHYID	4 bits	Zero-based PHY Identifier.
3 [25:16]	User Pattern 1	PAT1	10 bits	Two 10-bit user patterns must be provided by user, e.g. 0x0309, 0x02FC. Any value can be used.
3 [9:0]	User Pattern 2	PAT2	10 bits	Two 10-bit user patterns must be provided by user, e.g. 0x0309, 0x02FC. Any value can be used.
4 [9:0]	Threshold	THRSHLD	10 bits	Threshold value for the disparity or code violation error count. If the error count exceeds this value, it will trigger an event that can be reported by the corresponding threshold report.
5 [25:16]	Code Violation Pattern	CODE_VIOL_PAT	10 bits	User must specify a code violation pattern that is inserted. The pattern can have any 10-bit value, e.g. 0x032C.
5 [9:0]	PRBS Error Mask	PRBS_ERR_MSK	10 bits	A 10-bit mask to change the generated PRBS to induce errors before it is sent, e.g. 0x0303 can change up to 4 bits based on the generated pattern.
6	Performance Monitor	PMON	32 bits	A period that is used with THRSHLD field. After this period, the error counters would be reset. Unit is 10 ms. Valid range is from 0x00 to 0xFFFFFFF. 0x00 and 0xFFFFFFF disable the monitor timer.
7	PERF 1 Counter Control	PERF1CTL	32 bits	<p>A 32-bit field used to control the performance counters.</p> <p>Bit 31, when set to 1, triggers all the performance counters to update.</p> <p>Bit 30 is reserved.</p> <p>Bits [29:25] are reserved.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description																																																								
				<p>Bits [24:20] set to "Value" as defined in the table below to count the associated event.</p> <table> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>0</td><td>An SMP connection has reached the maximum allowed connection time.</td></tr> <tr><td>1</td><td>An STP connection has reached the maximum allowed connection time.</td></tr> <tr><td>2</td><td>An SSP connection has reached the maximum allowed connection time.</td></tr> <tr><td>3</td><td>We transmitted SATA_R_ERR.</td></tr> <tr><td>4</td><td>We received SATA_R_ERR.</td></tr> <tr><td>5</td><td>We received a good STP frame.</td></tr> <tr><td>6</td><td>We transmitted an STP frame.</td></tr> <tr><td>7</td><td>We received a bad SMP frame and broke the connection.</td></tr> <tr><td>8</td><td>We transmitted an SMP frame and received a break response.</td></tr> <tr><td>9</td><td>We received a good SMP frame.</td></tr> <tr><td>10</td><td>We transmitted a good SMP frame.</td></tr> <tr><td>11</td><td>We transmitted a NAK (CRC_ERROR).</td></tr> <tr><td>12</td><td>We received a NAK (CRC_ERROR) or an ACK/NAK timeout occurred.</td></tr> <tr><td>13</td><td>We received a good SSP frame.</td></tr> <tr><td>14</td><td>We transmitted an SSP frame.</td></tr> <tr><td>15</td><td>We detected a CRC error in a received data frame.</td></tr> <tr><td>16</td><td>We detected an error in the received test pattern.</td></tr> <tr><td>17</td><td>A break timeout occurred.</td></tr> <tr><td>18</td><td>A connection was established.</td></tr> <tr><td>19</td><td>AIP(WAITING_ON_CONNECTION) was received.</td></tr> <tr><td>20</td><td>AIP (WAITING_ON_PARTIAL) was received.</td></tr> <tr><td>21</td><td>We transmitted CREDIT_BLOCKED.</td></tr> <tr><td>22</td><td>We received CREDIT_BLOCKED.</td></tr> <tr><td>23</td><td>We transmitted a retry class open reject.</td></tr> <tr><td>24</td><td>We received a retry class open reject.</td></tr> <tr><td>25</td><td>We transmitted an abandon class open reject.</td></tr> <tr><td>26</td><td>We received an abandon class open reject.</td></tr> </tbody> </table>	Value	Definition	0	An SMP connection has reached the maximum allowed connection time.	1	An STP connection has reached the maximum allowed connection time.	2	An SSP connection has reached the maximum allowed connection time.	3	We transmitted SATA_R_ERR.	4	We received SATA_R_ERR.	5	We received a good STP frame.	6	We transmitted an STP frame.	7	We received a bad SMP frame and broke the connection.	8	We transmitted an SMP frame and received a break response.	9	We received a good SMP frame.	10	We transmitted a good SMP frame.	11	We transmitted a NAK (CRC_ERROR).	12	We received a NAK (CRC_ERROR) or an ACK/NAK timeout occurred.	13	We received a good SSP frame.	14	We transmitted an SSP frame.	15	We detected a CRC error in a received data frame.	16	We detected an error in the received test pattern.	17	A break timeout occurred.	18	A connection was established.	19	AIP(WAITING_ON_CONNECTION) was received.	20	AIP (WAITING_ON_PARTIAL) was received.	21	We transmitted CREDIT_BLOCKED.	22	We received CREDIT_BLOCKED.	23	We transmitted a retry class open reject.	24	We received a retry class open reject.	25	We transmitted an abandon class open reject.	26	We received an abandon class open reject.
Value	Definition																																																											
0	An SMP connection has reached the maximum allowed connection time.																																																											
1	An STP connection has reached the maximum allowed connection time.																																																											
2	An SSP connection has reached the maximum allowed connection time.																																																											
3	We transmitted SATA_R_ERR.																																																											
4	We received SATA_R_ERR.																																																											
5	We received a good STP frame.																																																											
6	We transmitted an STP frame.																																																											
7	We received a bad SMP frame and broke the connection.																																																											
8	We transmitted an SMP frame and received a break response.																																																											
9	We received a good SMP frame.																																																											
10	We transmitted a good SMP frame.																																																											
11	We transmitted a NAK (CRC_ERROR).																																																											
12	We received a NAK (CRC_ERROR) or an ACK/NAK timeout occurred.																																																											
13	We received a good SSP frame.																																																											
14	We transmitted an SSP frame.																																																											
15	We detected a CRC error in a received data frame.																																																											
16	We detected an error in the received test pattern.																																																											
17	A break timeout occurred.																																																											
18	A connection was established.																																																											
19	AIP(WAITING_ON_CONNECTION) was received.																																																											
20	AIP (WAITING_ON_PARTIAL) was received.																																																											
21	We transmitted CREDIT_BLOCKED.																																																											
22	We received CREDIT_BLOCKED.																																																											
23	We transmitted a retry class open reject.																																																											
24	We received a retry class open reject.																																																											
25	We transmitted an abandon class open reject.																																																											
26	We received an abandon class open reject.																																																											

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>27 We initiated a break.</p> <p>28 We received a break.</p> <p>29 We detected a CRC error on a received address frame.</p> <p>30 We detected a SAS error primitive</p> <p>Others Reserved</p> <p>Bits [19:0] are reserved.</p>

Table 94 SAS Diagnostic Command Type and Command Description Valid Combination

Command Type (CMD_TYPE)	Command Descriptor (CMD_DESC)	Resulting Operation
DIAG_OPRN_PERFORM	PRBS	Start inserting PRBS patterns.
	CJTPAT	Start inserting CJTPAT patterns.
	USR_PATTERNS	Start inserting User-defined patterns.
	PRBS_ERR_INSERT	Insert Erroneous PRBS pattern once.
	PRBS_INVERT	Start inserting inverted PRBS patterns.
	CJTPAT_INVERT	Invert CJTPAT pattern once.
	CODE_VIOL_INSERT	Insert code violation once.
	DISP_ERR_INSERT	Insert disparity error once.
	LINE_SIDE_ANA_LPBK	Configure line side analog loopback. See Figure 49 .
	LINE_SIDE_DIG_LPBK	Configure line side digital loopback.
DIAG_OPRN_STOP	SYS_SIDE_ANA_LPBK	Configure system side analog loopback. See Figure 49 .
	PRBS	Stop inserting and receiving PRBS patterns. See Figure 49 .
	CJTPAT	Stop inserting and receiving CJTPAT patterns.
	USR_PATTERNS	Stop inserting and receiving user-defined patterns.
	LINE_SIDE_ANA_LPBK	Unconfigure line-side analog loopback. See Figure 49 .
	LINE_SIDE_DIG_LPBK	Unconfigure line-side digital loopback. See Figure 49 .
RECEIVE_ENABLE	SYS_SIDE_ANA_LPBK	Unconfigure system-side analog loopback. See Figure 49 .
	PRBS	Enable receiving and processing PRBS.
CJTPAT	CJTPAT	Enable receiving and processing CJTPATs.
THRESHOLD_SPECIFY	CODE_VIOL_INSERT	Specify code violation error interval threshold.

Command Type (CMD_TYPE)	Command Descriptor (CMD_DESC)	Resulting Operation
	DISP_ERR_INSERT	Specify disparity error interval threshold.
DIAG_REPORT_GET	PRBS_ERR_CNT	Get PRBS error count report.
	CODE_VIOL_ERR_CNT	Get code violation error count report.
	DISP_ERR_CNT	Get disparity error count report.
	SSPA_PERF_CNT	Get error count of SSPA PERF1 Count register.
	PHY_RST_CNT	PHY reset failed count.
	LOST_DWD_SYNC_CNT	Get lost DWord sync count report.
	INVALID_DWD_CNT	Get invalid DWord count report.
	CODE_VIOL_ERR_THHD	Get code violation error interval threshold information report.
	DISP_ERR_THHD	Get disparity error interval threshold information report.
ERR_CNT_RESET	PRBS_ERR_THHD	Reset PRBS error threshold.
	PRBS_ERR_CNT	Reset PRBS error count.
	CODE_VIOL_ERR_CNT	Reset code violation error count.
	DISP_ERR_CNT	Reset disparity error count.
	SSPA_PERF_CNT	Reset error count of SSPA Perf1 Count register.
	LOST_DWD_SYNC_CNT	Reset lost DWord sync count.
	INVALID_DWD_CNT	Reset invalid DWord count.
	CODE_VIOL_ERR_THHD	Reset code violation error interval threshold information.
	DISP_ERR_THHD	Reset disparity error interval threshold information.
	PRBS_ERR_CNT_THHD	Reset PRBS error count and PRBS error threshold information.
	CODE_VIOL_ERR_CNT_THHD	Reset code violation error count and code violation error interval threshold information.
	DISP_ERR_CNT_THHD	Reset disparity error count and disparity error interval threshold Information.
	ALL	Reset all count and threshold information register bits.

Table 95 Command Descriptors for Command Types DIAG_OPRN_PERFORM, DIAG_OPRN_STOP, THRESHOLD_SPECIFY and RECEIVE_ENABLE

Command Descriptor	Value	Description
PRBS	0x00	PRBS patterns

Command Descriptor	Value	Description
CJTPAT	0x01	CJTPATs
USR_PATTERNS	0x02	User-defined diagnostic patterns
PRBS_ERR_INSERT	0x08	Erroneous PRBS induced using PRBS error mask
PRBS_INVERT	0x09	Inverted PRBS patterns
CJTPAT_INVERT	0x0A	Inverted CJTPATs
CODE_VIOL_INSERT	0x0B	Code violation errors
DISP_ERR_INSERT	0x0C	Disparity errors
LINE_SIDE_ANA_LPBK	0x10	Line side analog loopback
LINE_SIDE_DIG_LPBK	0x11	Line side digital loopback
SYS_SIDE_ANA_LPBK	0x12	System side analog loopback

Table 96 Command Descriptors for Command Types DIAG_REPORT_GET and ERR_CNT_RESET

Command Descriptor	Value	Description
PRBS_ERR_CNT	0x00	PRBS error count
CODE_VIOL_ERR_CNT	0x01	Code violation error count
DISP_ERR_CNT	0x02	Disparity error count
LOST_DWD_SYNC_CNT	0x05	Lost DWord sync count
INVALID_DWD_CNT	0x06	Invalid DWord count
PRBS_ERR_CNT_THHD	0x08	PRBS error count and threshold comparison
CODE_VIOL_ERR_CNT_THHD	0x09	Code violation error count and threshold comparison
DISP_ERR_CNT_THHD	0x0A	Disparity error count and threshold comparison
SSPA_PERF_CNT	0x0B	Error count from the SSPA Performance Count register
PHY_RST_CNT	0x0C	PHY reset failed count
PRBS_ERR_THHD	0x18	PRBS error threshold
CODE_VIOL_ERR_THHD	0x19	Code violation error interval threshold
DISP_ERR_THHD	0x1A	Disparity error interval threshold
ALL	0x1F	All count and threshold information register bits.

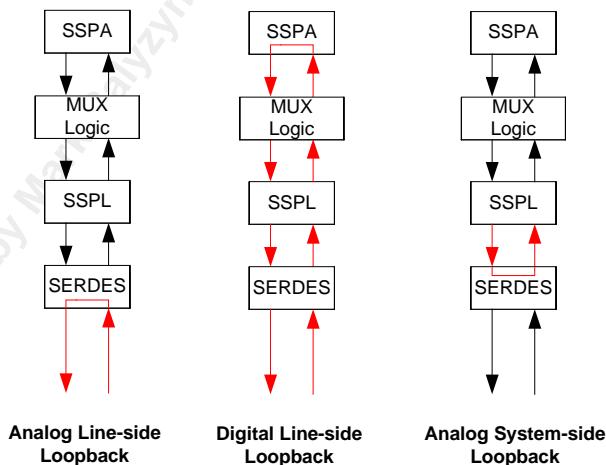
Table 97 SAS Diagnostic Command — Required Argument

Command	Argument	Size in Bits	Description
Start inserting user-defined patterns	User Pattern 1 (UT_PAT1)	10	Two 10-bit user patterns must be provided by user, e.g. 0x0309, 0x02FC. Any value can be used.
	User Pattern 2 (UT_PAT2)	10	
Insert erroneous PRBS pattern once	PRBS Error Mask (PRBS_ERR_MSK)	10	A 10-bit mask to change the generated PRBS to induce errors before it is sent, e.g. 0x0303 can change up to 4 bits based on what pattern is generated.
Insert code violation	Code Violation Pattern	10	User has to specify a code violation pattern that is inserted. It can have any 10-bit value,

Command	Argument	Size in Bits	Description
once	(CODE_VIOL_PAT)		e.g. 0x032C.
Specify PRBS error threshold	Threshold (THRSHLD)	8	The PRBS error threshold specifies the tolerance before loss of PRBS synchronization is declared because number of PRBS errors exceeding the threshold, e.g. 0x2A.
Specify code violation error interval threshold	Threshold (THRSHLD)	8	The code violation interval threshold parameter specifies the number of allowable code violation errors that can be tolerated within the PMON period.
Specify disparity error interval threshold	Threshold (THRSHLD)	8	The disparity error interval threshold parameter specifies the number of allowable disparity errors that can be tolerated within the PMON period.
Get error count of SSPA Performance Count register	Control (PERF1CTL)	32	A 32-bit field used to control the performance counters. See Table 93 for the definition of PERF1CTL. For example, to monitor or read the CJTPAT pattern error count, set PERF1CTL to 0x81000000.

The different loopbacks are shown in the figure below.

Figure 49 Line-side and System-side Loopbacks



The three different loopback modes are shown in [Figure 49](#). In this figure each of the PHYs consist of four components: SERDES, SSPL, Mux Logic, and SSPA. For the Analog line-side loopback configuration, the PHY's SERDES component receives the analog signal on the receive line (RX). Within the SERDES component the signal is "looped" over to the transmit (TX) line for external transmit. The red arrows indicate the path for each loopback.

The Digital line-side loopback allows the analog signal received (RX) at the SERDES component to continue through the digital components, where at the SSPA component, the signal is "looped" back. For both Analog and Digital line-side loopback modes, the signal received at (RXx) is from an external source (i.e., another PHY) and after loopback will be transmitted externally on TX.

For Analog System-side loopback configuration, the signal is generated from the SSPA component within the PHY on the TX line. Once at the SERDES component, the signal is "looped" over to the RX and fed back through to the digital components. For this loopback mode, the signal generated and received remain within the PHY.

A typical SAS diagnostic process involving only the SPC 8x6G is shown as in the following procedure (the square bracket implies commands that you can use as options for your required loopback):

1. Start the PHY(s) in diagnostic mode.
2. [Set up loopback].
3. [Enable pattern reception].
4. [Start pattern insertion].
5. [Select SSPA event to be monitored. Example: CJTPAT error – PERF1CTL = 0x81000000].
6. Reset counters.
7. [Insert errors].
8. Get report of error counters (CJTPAT error – PERF1CTL = 0x81000000).
9. Check if the counter values are expected.
10. [Stop the patterns].
11. [Stop the loopback].
12. Exit diagnostic mode.

7.24 SAS_HW_EVENT_ACK Command

Description

This command is sent to acknowledge that the event is sent by the SPC 8x6G through a [SAS_HW_EVENT Notification](#), which is described in Section 8.2. The completion of this operation is reported using a [SAS_HW_EVENT_ACK Command](#).

The event acknowledgement is used:

- As an event and acknowledge handshake between the SPC 8x6G and the host, allowing the host to pace the rate of the event messages generated for the host.
- For state synchronization between the SPC 8x6G and the host, such as during a PHY down and during port invalidation when the last PHY in the port is down.

The event acknowledgement is used for the following events received in a [SAS_HW_EVENT Notification](#) (Section 8.2):

- IOP_EVENT_BROADCAST_CHANGE
- IOP_EVENT_PHY_DOWN (only on the last PHY down event to indicate the port release, reported through subsequent IOP_EVENT_PORT_RECOVERY_TIMER_TMO or IOP_EVENT_PORT_RESET_TIMER_TMO)
- IOP_EVENT_PHY_ERR_INVALID_DWORD
- IOP_EVENT_PHY_ERR_DISPARITY_ERROR.
- IOP_EVENT_PHY_ERR_CODE_VIOLATION
- IOP_EVENT_PHY_ERR_LOSS_OF_DWORD_SYNCH
- IOP_EVENT_PHY_ERR_PHY_RESET_FAILED
- IOP_EVENT_PHY_ERR_INBOUND_CRC

Note: Only on the last PHY down event to indicate port release does a PHY down event condition (IOP_EVENT_PHY_DOWN) require an acknowledgment, see Section 11.5.1, “[PHY Down Handler](#)” and Section 11.5.2, “[Local PHY Control Link/Hard Reset Handler \(Host Initiated\)](#)”.

Usage

Initiator and target.

IOMB Format

Table 98 SAS_HW_EVENT_ACK Command Format

	Byte 3				Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0025					
1	HTAG												
2	Reserved			SEA				PHYID	PORT_ID				
3	PARAM0												
4	PARAM1												
5	Reserved												
...													
15													

Table 99 SAS_HW_EVENT_ACK Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2[23:8]	Source of event to acknowledge	SEA	2 Bytes	<p>0x0009: IOP_EVENT_BROADCAST_CHANGE Command to acknowledge the IOP_EVENT_BROADCAST_CHANGE event received in the SAS_HW_EVENT Notification (Section 8.2) .</p> <p>Contents of PARAM0:</p> <p>Bit 0 0b Clear the additional BROADCAST CHANGE events that have been latched between the last IOP_EVENT_BROADCAST_CHANGE reported to the host and the time the SPC 8x6G received this acknowledgement. New BROADCAST CHANGE events will be reported when received after the SPC 8x6G receives this acknowledgement.</p> <p>1b Report a new event if there is additional BROADCAST CHANGE events that have been latched between the last IOP_EVENT_BROADCAST_CHANGE event reported to the host and the time the SPC 8x6G received this acknowledgement.</p> <p>Other Bits Reserved</p> <p>Other bits reserved.</p> <p>0x0007: IOP_EVENT_PHY_DOWN Command to acknowledge the IOP_EVENT_PHY_DOWN event received in the SAS_HW_EVENT Notification. This acknowledgement is only used for acknowledging a PHY down in the last PHY in the port, implicitly acknowledges that the SPC 8x6G can release the PORT_ID.</p> <p>Sending the PHY down acknowledgement through SAS_HW_EVENT_ACK Command in some cases is done after subsequent reception of other event in SAS_HW_EVENT Notification such as IOP_EVENT_PORT_RECOVERY_TIMER_TMO or IOP_EVENT_PORT_RESET_TIMER_TMO when the PS is becoming PORT_INVALID.</p> <p>See Section 11.5.1, “PHY Down Handler” and Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for details about PHY down events.</p> <p>Contents of PARAM0: Not used.</p> <p>0x0012: IOP_EVENT_PHY_ERR_INVALID_DWORD 0x0013: IOP_EVENT_PHY_ERR_DISPARITY_ERROR 0x0014: IOP_EVENT_PHY_ERR_CODE_VIOLATION 0x0015:</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				IOP_EVENT_PHY_ERR_LOSS_OF_DWORD_SYMBOL: 0x0016: IOP_EVENT_PHY_ERR_PHY_RESET_FAILED 0x00C: IOP_EVENT_PHY_ERR_INBOUND_CRC Commands to acknowledge the IOP_EVENT_PHY_ERR_XXX events received in the SAS_HW_EVENT Notification . This acknowledgement is used to pace the PHY error reporting. The SPC 8x6G will not send another PHY error notification until the previous notification is acknowledged by the host.
2 [7:4]	PHY Identifier	PHYID	4 bits	Zero-based PHY identifier. The PHY identifier where this event was originally received in the SAS_HW_EVENT Notification .
2 [3:0]	Port Identifier	PORT_ID	4 bits	The PORT_ID where this event was originally received in the SAS_HW_EVENT Notification .
3	ACK Parameter 0	PARAM0	4 Bytes	Event specific acknowledgment parameter.
4	ACK Parameter 1	PARAM1	4 Bytes	Event specific acknowledgment parameter. Currently unused.

7.25 GET_TIME_STAMP Command

Description

This command is sent to the SPC 8x6G controller to get the SPC 8x6G internal time stamp used for event logging. The time stamp is based on a 64-bit timer with a resolution of 8 nsec.

The response to this command is reported through a [GET_TIME_STAMP Response](#) described in Section [8.23](#).

Getting the SPC 8x6G time stamp is typically for debugging to associate certain events received on the host with the event log entries generated by the SPC 8x6G..

Usage

Initiator and target.

IOMB Format

Table 100 GET_TIME_STAMP Command Format

	Byte 3				Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0026			
1	HTAG										
2											
...	Reserved										
15											

Table 101 GET_TIME_STAMP Command Field

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.

7.26 PORT_CONTROL Command

Description

This command is sent to profile or manage the SPC 8x6G firmware local port specified by PORT_ID. The completion of this command is reported in the [PORT_CONTROL Response](#) described in Section [8.25](#).

This command is used to control the following port properties:

- Setting the number of PHYs in a wide port (of a specific PORT_ID) that can be used for SMP traffic.
- Setting the Port Recovery Time value for a specific PORT_ID.
- Aborting all I/Os in all DEVICE_IDS accessed through a specific PORT_ID.
- Setting the port reset time value for a specific PORT_ID.
- Initiating a local port hard reset sequence, that is, in response to receiving a hard reset on the port.

Usage

Initiator and target.

Command Format

Table 102 PORT_CONTROL Command Format

	Byte 3				Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0027					
1	HTAG												
2	Reserved				PORT_OP		Reserved	PORT_ID					
3	PARAM 0												
4	PARAM 1												
5	Reserved												
...	Reserved												
15													

Table 103 PORT_CONTROL Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation. 0x01: SET_SMP_PHY_WIDTH. Set the number of PHYs in a wide port that can be used for sending SMP requests. Value of PARAM 0: 0x0000: No limit. Use all possible PHYs in the wide port. 0x0001 to the number of PHYs in wide port: The limit of PHYs that can be used for sending SMP. Value of PARAM 1: Not used.
2 [8:15]	Port control operation	PORT_OP	1 Byte	0x02: SET_PORT_RECOVERY_TIME. Set the value for Port Recovery Time for this PORT_ID. See Section 11.5.1, “PHY Down Handler” for the description and use of Port Recovery Time. Value of PARAM 0: Time in 100 ms for the Port Recovery Time. A value of zero indicates that Port Recovery Time scheme is disabled. The value of PORT_RECOVERY_TIME should be greater than the maximum IT_NEXUS_LOSS time of the drive/target set for the devices accessed through a given port. Value of PARAM 1: Not used. 0x03: PORT_IO_ABORT. Abort/cleanup all I/Os in all DEVICE_IDs accessed through a specific PORT_ID.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>If the SPC 8x6G controller finds that there are pending I/Os associated with PORT_ID, for each pending I/O it will send a SSP_COMPLETION Response (Section 8.3), SATA_COMPLETION Response (Section 8.9) or SMP_COMPLETION Response (Section 8.4) with STATUS set to IO_ABORTED, finally followed by PORT_CONTROL Response (Section 8.25).</p> <p>Value of PARAM 0:</p> <p>Bit #0: Flag to indicate if I/O resource needs to be quarantined:</p> <ul style="list-style-type: none"> 0: Requires I/O resource quarantine. 1: Do not quarantine the I/O resource. <p>Value of PARAM 1: Not used.</p> <p>0x04: SET_PORT_RESET_TIME. Set the value for Port Reset Time for this PORT_ID. See Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for the Port Reset Time.</p> <p>Value of PARAM 0:</p> <p>Time in 100 ms for the Port Reset Time when used as SAS port. Minimum value is 300 ms.</p> <p>Value of PARAM 1:</p> <p>Time in 100 ms for the Port Reset Time when used as SATA port. Minimum value is TBD ms.</p> <p>0x05: HARD_RESET. To initiate several type of operations in sequence in error recovery such as in response to receiving hard reset on the port:</p> <ul style="list-style-type: none"> ○ Abort/cleanup all I/Os in all DEVICE_IDS belonging to the port. If the port has at least one initiator role defined in a PHYs ID frame, aborted I/Os will be quarantined. For a target port, it does not quarantine aborted I/Os. ○ If the bit 0 of PARAM0 is set, deregister all device handles (DEVICE_IDS) belonging to the port. Each device registration will result with a DEVICE_HANDLE_REMOVED Notification, which is described in Section 8.30. If the bit 0 of PARAM0 is not set, this step is skipped. ○ Perform local LINK RESET on all PHYs belonging to the port. <p>The response to this operation is different for narrow port and wide port. Also the response in narrow port is different depending on</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>whether Port Recovery Time is enabled or not. See Section 11.5.5, “Hard Reset Received Handler” for the use and sequence of this operation.</p> <p>Value of PARAM 0:</p> <p>Bit 0:</p> <ul style="list-style-type: none"> 1b: Automatically deregister devices as part of HARD_RESET operation. 0b: Skip deregister devices as part of the HARD_RESET operation. <p>Bits 31:1: Reserved.</p> <p>Value of PARAM 1: Not used</p>
2 [3:0]	Port Identifier	PORT_ID	4 bits	The ID for the SPC 8x6G controller's local SAS/SATA port context.
3	Parameter 0	PARAM 0	4 Bytes	Operation specific parameter.
4	Parameter 1	PARAM 1	4 Bytes	Operation specific parameter.

7.27 GET_NVMD_DATA Command

This command is sent to the SPC 8x6G controller to do the following:

1. Get the vital product data (VPD) from flash memory or EEPROM.
2. Get data from the TWI device.
3. Get the register dump information from flash memory. For details about register dumps, see Section 11.2.15, “Firmware Fatal Errors Register Dump”.
4. Get the event log from flash memory.
5. Read back the Expansion ROM image from flash memory.

The completion of this command is reported in the [GET_NVMD_DATA Response](#), which is described in Section 8.28.

Notes

- This GET_NVMD_DATA Command IOMB replaced the older GET_VPD Command IOMB and TWI Command IOMB.
- When accessing the TWI using NVMD = 0000b, the host is required to set the correct TWI parameters (such as address, length and offset). The SPC 8x6G cannot validate the TWI device it is accessing and must rely on valid parameters from the host.

7.27.1 Getting VPD Data

Customer-specific SPC 8x6G VPD information is stored either in flash memory or SEEPROM device(s).

The first 512 bytes of configuration SEEPROM is reserved for boot configuration and cannot be used for customer-specific VPD. See Section 4.3, “[Configuration SEEPROM](#)” for details of the configuration information for the configuration SEEPROM. Other SEEPROM device could be connected to the TWI bus and used for additional VPD storage.

The VPD in flash memory is defined by the SPC 8x6G firmware at a specified flash memory offset and partition. The maximum VPD data in flash memory is 4 Kbytes. See Section 4.5, “[Flash Memory Format](#)” for the flash memory mapping.

There are two modes where the VPD operation can be executed: Direct response in IOMB (up to 48 bytes) or indirect response using physical address fields. For reading from the VPD flash memory only indirect mode is supported.

7.27.2 Getting Data from a TWI Device

This command is used to manage the TWI device attached to the SPC 8x6G’s TWI. The TWI device includes (but not limited to) the SEEPROM device. When the NVM Device (NVMD) field is set to 0000b (TWI devices), this command also allows the host to specify the:

- TWI device address
- TWI bus number
- TWI device page size (1, 8, 16, or 32 bytes)
- TWI device address size (1 or 2 bytes)

For reading the TWI device (including SEEPROM) both direct and indirect mode are supported.

7.27.3 Getting Register Dump Information from Flash Memory

The register dump is written to the flash non-volatile memory (NVM) following the SPC 8x6G critical error.

Getting the SPC 8x6G register dump is typically done for debug purposes. For detailed information on critical error reporting, see Section 11.2, “[Device Specific Fatal Errors](#)”. For details about register dumps, see Section 11.2.15, “[Firmware Fatal Errors Register Dump](#)”.

There is one 16-Kbyte register dump partition in flash memory allocated for each of the SPC 8x6G internal CPUs. The host must access the register dump in flash memory using a 4-Kbyte chunk at a time.

Only indirect mode is supported when reading register dumps from flash memory.

This feature is only available when the SPC 8x6G firmware is loaded from flash memory or HDA mode where flash memory exists. See Section 3.21, “[Host Direct Access \(HDA\) Mode](#)” for details on HDA mode.

When flash firmware boot is not used or available, such as the case in HDA mode without flash memory, the host should access the register dump memory location directly from the GSM location specified by the DWord 0x1D, Fatal Error Register dump offset, and DWord 0x1F, Fatal Error Register dump offset for the IOP for the MSGU in the MPI Configuration Table (Main Part) as described in [Table 38](#).

7.27.4 Getting Event Log Information from Flash Memory

The event log is written to the flash non-volatile memory (NVM) following an SPC 8x6G critical error.

The SPC 8x6G event log is typically retrieved for debug purposes. For detailed information on critical error reporting, see Section 11.2, “[Device Specific Fatal Errors](#)”. For details about the event log, see Section 3.17, “[Event Log Operation](#)”.

There is one 16-Kbyte event log region (offset 0x1_0000) in partition 7 for the MSGU(AAP1) and partition 15 for the IOP in the flash memory map ([Table 35](#)). The host must access the event log in flash memory using one 4-Kbyte chunk at a time.

Only indirect mode is supported when reading register dumps from flash memory.

This feature is only available when the SPC 8x6G firmware is loaded from flash memory or in HDA mode where flash memory exists. See Section 3.21, “[Host Direct Access \(HDA\) Mode](#)” for details on HDA mode.

When flash firmware boot is not used or available, such as the case in HDA mode without flash memory, the host should access the register dump memory location directly from the GSM location specified by the DWord 0x14 and 0x15 for MSGU (AAP1) event log address, and DWord 0x18 and 0x19 for IOP event log address as described in [Table 38](#).

7.27.5 Reading Back The Expansion ROM Image from Flash Memory

This command allows reading back the Expansion ROM image previously written to the flash memory using [FW_FLASH_UPDATE Command](#). See Section 7.20 for the details of [FW_FLASH_UPDATE Command](#).

Usage

Initiator and target.

IOMB Format

Table 104 GET_NVMD Command Format

	Byte 3				Byte 2		Byte 1		Byte 0								
0	V	H	R	BC=1	Reserved	OBID	CAT=0x2	OPC=0x028									
1	HTAG																
2	IP	Reserved		TDA		TBN	TDPS	TDAS	NVMD								
3	D_LEN			DOA													
4	Reserved																
...																	
11																	
12	IPBAL																
13	IPBAH																
14	IPDL																
15	Reserved																

Table 105 GET_NVMD Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [31]	Indirect Payload	IP	1 bit	Flag indicating the data payload mode: 0b: Direct payload mode. Use the DWords [15:4] in outbound IOMB for the data response. 1b: Indirect payload mode. Use DWords [13:12] as the physical address for the data response buffer in host memory. Note: If NVMD = 0001b (Configuration EEPROM device), 0100b (VPD flash memory), 0101b (AAP1 register dump or event log flash memory), 0110b (IOP register dump or event log flash memory), and 0111b (Expansion ROM flash memory), only IP=1b mode is supported.
2 [23:16]	TWI Device Address	TDA	1 Byte	The address of the TWI device to read from. Valid if NVMD is set to 0000b.
2 [15:12]	TWI Bus Number	TBN	4 bits	TWI bus number. Valid only if NVMD = 0000b.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2 [11:8]	TWI Device Page Size	TDPS	4 bits	TWI Device Page Size. Valid only if NVMD = 0000b. 0000b: 1 byte. 0001b: 8 bytes. 0010b: 16 bytes. 0011b: 32 bytes. Others: Reserved.
2 [7:4]	TWI Device Address Size	TDAS	4 bits	TWI Device Address Size. Valid only if NVMD = 0000b. 0000b: 1 bytes. 0001b: 2 bytes. Others: Reserved.
2 [3:0]	NVM Device	NVMD	4 bits	Device to access: 0000b: TWI devices. 0001b: Specific Configuration SEEPROM device (first 512 bytes consist of Configuration SEEPROM). 0100b: VPD flash memory. 0101b: AAP1 register dump flash memory. 0110b: IOP register dump flash memory. 0111b: Expansion ROM flash memory. Others: Reserved. <p>Note: The Configuration SEEPROM can also be accessed by setting NVMD=0000b. However, the related TBN, TDPS, and TDAS fields must be set correctly. If the Configuration SEEPROM device is larger than 512 bytes, the additional space can be accessed in the same way.</p>
3 [31:24]	Direct Payload Data Length	D_LEN	1 Byte	Direct data payload length in bytes, valid only if IP is set to 0. The maximum length is 48 bytes.
3 [23:0]	Data Offset Address	DOA	3 Bytes	Offset address in the device for data. If NVMD is set to 0000b (TWI devices), 0100b (VPD flash memory), 0101b (AAP1 register dump or event log) or 0110b (IOP register dump or event log) this is the offset address for data in the device. Note: If the event log is accessed, the start offset address is 0x1_0000. If NVMD is set to 0001b, this field should be set to 0.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
12	Indirect Payload Buffer Address Lower	IPBAL	4 Bytes	The lower 32-bit physical address for the response buffer in host memory. Only valid if the IP bit is set to 1b.
13	Indirect Payload Buffer Address Higher	IPBAH	4 Bytes	The higher 32-bit physical address for the response buffer in host memory. Only valid if the IP bit is set to 1b.
14	Indirect Payload Data Length	IPDL	4 Bytes	The size of the host memory allocated for the response. The maximum length is 4 Kbytes. Only valid if the IP bit is set to 1b.

7.28 SET_NVMD_DATA Command

This command is sent to the SPC 8x6G controller to do the following:

1. Write the vital product data (VPD) to flash memory or SEEPPROM.
2. Write data to the TWI device.

The completion of this command is reported in the [SET_NVMD_DATA Response](#) described in Section [8.29](#).

Notes

- This SET_NVMD_DATA Command IOMB replaced the older SET_VPD Command IOMB and TWI Command IOMB.
- When accessing the TWI using NVMD = 0000b, the host is required to set the correct TWI parameters (such as address, length and offset). The SPC 8x6G cannot validate the TWI device it is accessing and must rely on valid parameters from the host.

Writing VPD Data

Customer-specific SPC 8x6G vital product data (VPD) information is stored either in flash memory or SEEPPROM device(s).

The first 512 bytes of the configuration SEEPPROM is reserved for boot configuration and cannot be used for customer-specific VPD. See Section [4.3, “Configuration SEEPPROM”](#) for details of the configuration information on the configuration SEEPPROM. Other SEEPPROM devices can be connected to the TWI bus and used for additional VPD storage.

The VPD in flash memory is defined by the SPC 8x6G firmware at a specified flash memory offset and partition. The maximum VPD data in flash memory is 4 Kbytes. See Section [4.5, “Flash Memory Format”](#) for the flash memory mapping.

There are two modes where the VPD operation could be executed: direct data in IOMB (up to 48 bytes) or indirect data using physical address fields. For writing to VPD flash memory only indirect mode is supported.

Writing Data to TWI Device

This command is used to manage the TWI device attached to the SPC 8x6G's TWI. The TWI device includes (but not limited to) the SEEPROM device. When NVMD field is set to 0000b (TWI devices), this command also allows the host to specify:

- TWI Device Address
- TWI Bus Number
- TWI Device Page Size (1, 8, 16, or 32 bytes)
- TWI Device Address Size (1 or 2 bytes)

For writing the TWI device (including SEEPROM) both direct and indirect mode are supported.

Usage

Initiator and target.

IOMB Format

Table 106 SET_NVMD Command Format

	Byte 3			Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1		Reserved	OBID	CAT=0x2	OPC=0x029					
1	HTAG													
2	IP	Reserved		TDA		TBN	TDPS	TDAS	NVMD					
3	D_LEN			DOA										
4	IPReserved / SIGNATURE / D_DATA[3:0]													
5	IPReserved / D_DATA[31:4]													
...														
11														
12	IPBAL / D_DATA[35:32]													
13	IPBAH / D_DATA[39:36]													
14	IPDL / D_DATA[43:40]													
15	IPReserved / D_DATA[47:44]													

Table 107 SET_NVMD Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [31]	Indirect Payload	IP	1 bit	<p>Flag indicating the data payload mode:</p> <p>0b: Direct payload mode. Use DWords [15:4] in the outbound IOMB for the data response.</p> <p>1b: Indirect payload mode. Use DWords [13:12] as the physical address for the data response buffer in host memory.</p> <p>Note: If NVMD = 0001b (Configuration EEPROM device) or 0100b (VPD flash memory), only IP=1b mode is supported.</p> <p>Note: The following limitations apply for NVMD = 0100b (VPD FLASH):</p> <ul style="list-style-type: none"> Send all of the data to be written to flash memory (not a subset of the data). First read the VPD data via the GET_NVMD_DATA Command (Section 7.27), modify the data, and then write the data via SET_NVMD_DATA Command. Direct mode is not supported as the size of the data can be up to 4K. Only indirect mode is supported. The offset field must be set to logic 0 as the complete VPD data is written all at once. Data will not be written to flash memory if a non-zero offset is used since these are not applicable.
2 [23:16]	TWI Device Address	TDA	1 Byte	The address of the TWI device to write to. Valid if NVMD is set to 0000b.
2 [15:12]	TWI Bus Number	TBN	4 bits	TWI bus number. Valid only if NVMD = 0000b.
2 [11:8]	TWI Device Page Size	TDPS	4 bits	<p>TWI Device Page Size.</p> <p>Valid only if NVMD = 0000b.</p> <p>0000b: 1 byte.</p> <p>0001b: 8 bytes.</p> <p>0010b: 16 bytes.</p> <p>0011b: 32 bytes.</p> <p>Others Reserved.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2 [7:4]	TWI Device Address Size	TDAS	4 bits	<p>TWI Device Address Size.</p> <p>Valid only if NVMD = 0000b.</p> <p>0000b: 1 bytes.</p> <p>0001b: 2 bytes.</p> <p>Others Reserved.</p>
2 [3:0]	NVM Device	NVMD	4 bits	<p>Device to access:</p> <p>0000b: TWI devices.</p> <p>0001b: Specific Configuration SEEPPROM device (first 512 bytes consist of Configuration SEEPPROM).</p> <p>0100b: VPD flash memory.</p> <p>Others: Reserved.</p> <p>Note: The Configuration SEEPPROM can also be programmed by setting NVMD=0000b. However, the related TBN, TDPS, TDAS and SIGNATURE fields must be set correctly and only indirect mode is supported (IP=1b). If the Configuration SEEPPROM device is larger than 512 bytes, the additional space can be programmed in the same way except there is no need to set the SIGNATURE field.</p>
3 [31:24]	Direct Payload Data Length	D_LEN	1 Byte	Direct data payload length in bytes, valid only if IP is set to 0. The maximum length is 48 bytes.
3[23:0]	Data Offset Address	DOA	3 Bytes	<p>Offset address in the device for data</p> <p>If NVMD is set to 0000b, this is the offset address for data in the TWI device.</p> <p>If NVMD is set to 0001b and 0100b (VPD flash memory), this field should be set to 0.</p>
4	Reserved / SIGNATURE / Direct Payload Data	IPReserved / SIGNATURE / D_DATA[3:0]	4 Bytes	<p>If IP=1b and NVMD=(0000b or 0100b), this is reserved for indirect payload mode.</p> <p>If IP=1b and NVMD=0001b, this is the SIGNATURE (0xFEDCBA98) for writing the configuration SEEPPROM.</p> <p>Note: if NVMD=0000b is used to access the Configuration SEEPPROM, the SIGNATURE (0xFEDCBA98) must be set.</p> <p>If IP=0b, this is the direct payload data bytes [3:0].</p>
11:5	Reserved / Direct Payload Data	IPReserved / D_DATA[31:4]	28 Bytes	<p>If IP=1b, this is reserved for indirect payload mode.</p> <p>If IP=0b, this is the direct payload data bytes [31:4].</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
12	Indirect Payload Buffer Address Lower / Direct Payload Data	IPBAL / DATA[35:32]	4 Bytes	If IP=1b, this is the lower 32-bit physical address for the response buffer in host memory. If IP=0b, this is the direct payload data bytes [35:32].
13	Indirect Payload Buffer Address Higher / Direct Payload Data	IPBAH / DATA[39:36]	4 Bytes	If IP=1b, this is the higher 32-bit physical address for the response buffer in host memory. If IP=0b, this is the direct payload data bytes [39:36].
14	Indirect Payload Data Length / Direct Payload Data	IPDL / DATA[43:40]	4 Bytes	If IP=1b, this is the size of the indirect payload mode data. The maximum length is 4 Kbytes. If IP=0b, this is the direct payload data bytes [43:40].
15	Reserved / Direct Payload Data	IPReserved / D_DATA[47:44]	4 Bytes	If IP=1b, this field is reserved. If IP=0b, this is the direct payload data bytes [47:44].

7.29 SET_DEVICE_STATE Command

Description

This command is sent to change the SPC 8x6G internal device state associated with a DEVICE_ID. The internal device state is used to control the flow of I/Os in the error recovery path. The definitions of the device states are provided in [Table 8](#) in Section 3.2, "Device Handle and DEVICE_ID".

Some of the error recovery paths that may require the use of the [SET_DEVICE_STATE Command](#) IOMB include:

- Device reset: If a device reset, issued by a HARD_RESET PHY control to the device, is used as part of error recovery, the [SET_DEVICE_STATE Command](#) can be used to prevent I/O requests that are already in the IQ to the target device from being sent.
- A special task management case: There is an option during an initiator-mode task management operation to prevent/block normal I/O (non-task management operation) from going to the target device by setting the SPC 8x6G internal device state. In this case, the host sets the device state to DS_IN_RECOVERY either implicitly as part of the [SSP_INI_TM_START Command](#) described in Section 7.5 or explicitly by calling [SET_DEVICE_STATE Command](#). See Section 11.5.6.2, "Task Management Special Control with Device State" for details on controlling the device state option for task management.

- A head-of-line blocking issue during an IT_NEXUS_LOSS timeout for an SSP device: Head-of-line blocking can occur in the SPC 8x6G when an SSP target is not responding in a timely manner (less than the default 2 seconds for IT_NEXUS_LOSS timeout), such as when the target device is missing. In a wide port, the non-progressing I/O will cause head-of-line blocking to the rest of I/Os already in the SPC8x6G FIFO queue destined to different target devices. The SPC 8x6G provides the [SET_DEVICE_STATE Command](#) to flush the I/Os pending in the per port I/Os FIFO queue associated with a particular DEVICE_ID that caused the head-of-line blocking, allowing other I/Os destined to different DEVICE_IDS to continue. For more details about error handling for SSP head-of-line blocking issues caused by IT_NEXUS_LOSS timeouts, see Section 11.5.6.3, “[Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Device](#)”.
- A head-of-line blocking issue for the SATA protocol: This can occur when the SATA drive does not respond because the SPC 8x6G is in an interlock state with it. For more details about error handling for SATA head-of-line blocking, see Section 11.5.6.4, “[Head-Of-Line Blocking for SATA Protocol](#)”.

The completion of this command is reported in the [SET_DEVICE_STATE Response](#), described in Section 8.31.

Usage

Initiator and target.

Command Format

Table 108 SET_DEVICE_STATE Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x002A
1	HTAG							
2	DEVICE_ID							
3	Reserved							NDS
4	Reserved							
...								
15								

Table 109 SET_DEVICE_STATE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation. This HTAG is only to be used to communicate between a host and the SPC 8x6G.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	Device Identifier	DEVICE_ID	4 Bytes	SPC device identifier. See Section 3.2, “Device Handle and DEVICE_ID” for a detailed description of DEVICE_ID.
3 [3:0]	New Device State	NDS	4 bits	New state to be set for the device. Valid choices are: 0x1 : DS_OPERATIONAL 0x3 : DS_IN_RECOVERY

7.30 GET_DEVICE_STATE Command

Description

This command is sent to get the SPC 8x6G internal device state associated with a DEVICE_ID.

See Table 8 in Section 3.2, “Device Handle and DEVICE_ID” for the definitions of device states.

The completion of this command is reported in the GET_DEVICE_STATE Response, described in Section 8.32.

Usage

Initiator and target.

Command Format

Table 110 GET_DEVICE_STATE Command Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x002B
1	HTAG							
2	DEVICE_ID							
3	Reserved							
...								
15								

Table 111 GET_DEVICE_STATE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation. This HTAG is only to be used to communicate between a host and the SPC 8x6G.
2	Device Identifier	DEVICE_ID	4 Bytes	SPC device identifier. See Section 3.2, “ Device Handle and DEVICE_ID ” for a detailed description of DEVICE_ID.

7.31 SET_DEVICE_INFO Command

Description

This command is sent to change some of the device parameters associated with a DEVICE_ID.

The completion of this command is reported in the [SET_DEVICE_INFO Response](#), as described in Section 8.33.

Usage

Initiator and target.

Command Format

Table 112 SET_DEVICE_INFO Command Format

	Byte 3			Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x002C				
1	HTAG											
2	DEVICE_ID											
3	Reserved				SA SR SI							
4	Reserved			A R	IT Nexus Timeout (ITNT)							
5	Reserved											
...												
15												

Table 113 SET_DEVICE_INFO Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2	Device Identifier	DEVICE_ID	4 Bytes	SATA or SSP device identifier. See Section 3.2, “Device Handle and DEVICE_ID” for a detailed description of DEVICE_ID.
3 [31:3]	Reserved			Reserved
3 [2]	Set AWT	SA	1 bit	Flag to indicate if a change to the AWT (Arbitration Wait Time) flag is requested: 0: Do not set the AWT flag with new value in 'A' field. 1: Set the AWT flag with new value in the 'A' field.
3 [1]	Set Retry Flag	SR	1 bit	Flag to indicate if change to the Transport Layer Retry (TLR) flag per the SAS 1.1 and SAS 2.0 specification is requested: 0: Do not set the Retry flag with a new value in the 'R' field. 1: Set the Retry flag with a new value in the 'R' field. See Section 3.20, “Transport Layer Retry (TLR) Handling”.
3 [0]	Set ITNT	SI	1 bit	Flag to indicate if a change to the IT Nexus Timeout (ITNT) is requested: 0: Do not set ITNT with new value in ITNT field. 1: Set ITNT with the new value in ITNT field.
4 [17]	AWT flag	A	1 bit	Valid if SA field is set to 01b. Priority setting for the Arbitration Wait Time (AWT) for this remote device: 0b: Default setting (recommended). The actual AWT value is based on how long an OPEN frame has been waiting for a connection request to be accepted. It starts at 0. As specified in the SAS specification, from 0 to 32768 µs, the AWT value is incremented every µs. From 32768 µs and on, the AWT value is incremented every ms. 1b: Increase priority. The actual AWT value starts at 32768 µs when the A flag is set to '1'. The AWT value is incremented every ms from that point on.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [16]	Retry flag	R	1 bit	<p>Valid if the SR field is set to 01b.</p> <p>This field enables the Transport Layer Retry (TLR) flag per the SAS 1.1 and SAS 2.0 specifications:</p> <p>1b: Enable TLR. 0b: Disable TLR.</p> <p>PMC-Sierra recommends that TLR is enabled (set to 1b).</p> <p>See Section 3.20, “Transport Layer Retry (TLR) Handling”.</p>
4 [15:0]	ITNT	ITNT	2 Bytes	<p>Valid if the SI field is set to 01b.</p> <p>The value in milliseconds of the time unit that is used by the SPC 8x6G to determine the nexus timeout condition.</p>

7.32 SAS_RE_INITIALIZATION Command

Description

This is an optional command used to change the SPC 8x6G's default internal SAS/SATA module resource allocation and internal SAS/SATA parameters. When used, this command is sent prior to any SAS IOMB operation such as a [PHY_START Command](#).

The completion of this command is reported in the [SAS_RE_INITIALIZATION Response](#) as described in Section [8.34](#).

Note: If this command is sent after other IOMBs other than the ECHO Command are executed then the device may exhibit unknown behavior.

Usage

Initiator and target.

Command Format

Table 114 SAS_RE_INITIALIZATION Command Format

	Byte 3			Byte 2		Byte 1		Byte 0							
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02							
1	HTAG														
2	Reserved	S M P O R R C	S O R R D	S O R N D O	S A H O L T	Reserved									
3	Reserved				Reserved	MAX_PORTS									
4	OPEN_REJECT_RETRIES_CMD				OPEN_REJECT_RETRIES_DATA										
5	Reserved				SATA_HOL_TMO										
6	Reserved														
...															
15															

Table 115 SAS_RE_INITIALIZATION Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [31]	Reserved		1 bit	Reserved
2 [30]	Reserved		1 bit	Reserved
2 [29]	Set MAX PORTS	SMPORT	1 bit	Flag to indicate if a change to the default SAS/SATA ports is requested: 0: Do not change the default setting number of ports. Default is 8 ports. 1: Set the number of ports according to the value in MAX_PORTS field.
2 [28]	Set OPEN REJECT RETRIES CMD	SORRC	1 bit	Flag to indicate if a change to the OPEN REJECT (RETRY) in the command phase is requested: 0: Do not change the default setting of OPEN REJECT (RETRY) in the command phase. Default is 256 retries. 1: Set the OPEN REJECT (RETRY) in the Command phase according to the count in OPEN_REJECT_RETries_CMD field.
2 [27]	Set OPEN REJECT RETRIES DATA	SORRD	1 bit	Flag to indicate if a change to the OPEN REJECT (RETRY) in the data phase is requested: 0: Do not change the default setting of OPEN REJECT (RETRY) in the data phase. Default is 2048 retries. 1: Set the OPEN REJECT (RETRY) in the data phase according to the count in OPEN_REJECT_RETries_DATA field.
2 [26]	Set OPEN REJECT NO DESTINATION OPTION	SORNDO	1 bit	Flag to indicate if an OPEN REJECT (NO DESTINATION) will be mapped into an OPEN REJECT (BAD DESTINATION). The default is not to do the mapping. 0: Do not remap. 1: Remap OPEN REJECT (NO DESTINATION) into OPEN REJECT (BAD DESTINATION).
2 [25]	Set SATA HOL TMO	SSAHOLT	1 bit	Flag to indicate if a change to the default delay for a SATA Head-of-Line Blocking detection timeout is requested: 0: Do not change the default setting of the SATA HOL Blocking detection timeout. The default is 500 milliseconds. 1: Set the SATA HOL Blocking timeout according to the value in the SATA_HOL_TMO field in 100 millisecond units.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2 [24:0]	Reserved			Reserved
3 [31:16]	Reserved		2 Bytes	Reserved
3 [15:8]	Reserved		2 Bytes	Reserved
3 [7:0]	Maximum SAS Ports	MAX_PORTS	1 Byte	This field is valid only if the SMPORT field is set to 1. Number of SAS ports supported. Should not exceed the maximum of 8 ports.
4 [31:16]	OPEN REJECT (RETRY) Command Phase	OPEN_REJECT_TRIES_CMD	2 Bytes	This field is valid only if the SORRC field is set to 1. Number of supported retries for an OPEN REJECT (RETRY) during the command phase (Initiator Mode) and Response Phase (Target Mode). The number should not exceed the maximum of 255 retries. One unit corresponds to 16 retries.
4 [15:0]	OPEN REJECT (RETRY) Data Phase	OPEN_REJECT_TRIES_DATA	2 Bytes	This field is valid only if the SORRD field is set to 1. Number of supported retries for an OPEN REJECT (RETRY) during the data phase (Initiator Mode and Target Mode) and XFER_RDY Phase (Target Mode). This number should not exceed the maximum of 255 retries. One unit corresponds to 16 retries.
5 [15:0]	SATA Head-Of-Line Blocking Detection Timeout	SATA_HOL_TO	2 Bytes	This field is valid only if the SSAT field is set to 1. The timeout delay used to detect the SATA Head-Of-Line Blocking. The delay is in 100 millisecond units. A zero delay means detection has been disabled.
5 [31:16]	Reserved		2 Bytes	Reserved

7.33 SGPIO_REGISTER Command

Description

The SGPIO_REGISTER command controls SGPIO LEDs on the enclosure of locally-attached drives only. It reads from/writes to SGPIO registers and sets one or more SGPIO registers. The register settings are based on the *SFF 8485 Specification for Serial GPIO(SGPIO) Bus*. These settings generate different output patterns. The SGPIO initiator includes A and B blink generators to control the output patterns. The SGPIO initiator operates in the normal mode. To indicate usage of this IOMB to SPC8x6G a bit needs to be set in the MPI Main Configuration Table. If this bit is set to 1, the host still needs to set the GPIO Enable bit by writing to the SGPIO_CFG[0] Configuration register defined in the SFF 8489 specification before sending other SGPIO IOMBs.

Normal mode is drive-by-drive access to the SDataOut and SDataIn bit streams. It uses a hardware-based activity tracker and blink generators.

The completion of this command is reported in the SGPIO Response. See section [8.35](#).

Usage

Initiator.

Command Format

Table 116 SGPIO_REGISTER Command Format

	Byte 3			Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x002E						
1	HTAG													
2	SgpioRegIndex		SgpioRegType		SgpioFunction		SgpioSMPFrameType							
3	Reserved3		Reserved2		Reserved1		SgpioRegCount							
4	Payload													
5														
...														
15														

Table 117 SGPIO_REGISTER Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [7:0]	SgpioSMPFrameType	SSFT	1 Byte	Set this field to 0x40. Refer to the SFF 8485 Specification for Serial GPIO (SGPIO) Bus.
2 [15:8]	SgpioFunction	SF	1 Byte	Set this field to either: 0x02: READ SGPIO REGISTER 0x82: WRITE SGPIO REGISTER Refer to the <i>SFF 8485 Specification for Serial GPIO (SGPIO) Bus</i> .
2 [23:16]	SgpioRegType	SRT	1 Byte	This field specifies the bank of registers to read/write. The SPC supports five register types per the <i>SFF 8485 Specification for Serial GPIO (SGPIO) Bus</i> . SGPIO_CONFIG_REG = 0x0 SGPIO_DRIVE_BY_DRIVE_RECEIVE_REG=0x1 SGPIO_GENERAL_PURPOSE_RECEIVE_REG=0x2 SGPIO_DRIVE_BY_DRIVE_TRANSMIT_REG =0x3, SGPIO_GENERAL_PURPOSE_TRANSMIT_REG=0x4 Refer to the <i>SFF 8485 Specification for Serial GPIO (SGPIO) Bus</i> .
2 [31:24]	SgpioRegIndex	SRI	1 Byte	This field specifies the index of the first register in the bank to read/write. Two registers for each register type are supported. SgpioRegIndex = 0x0 SgpioRegIndex = 0x1 Refer to the <i>SFF 8485 Specification for Serial GPIO (SGPIO) Bus</i> .
3 [7:0]	SgpioRegCount	SRC	1 Byte	This field specifies the number of registers starting with the specified index to read/write. Each register type supports a bank of two registers. Based on SgpioRegIndex, SgpioRegCount can be either: SgpioRegCount=0x1 SgpioRegCount=0x2 Refer to the <i>SFF 8485 Specification for Serial GPIO (SGPIO) Bus</i> .
3 [31:8]	Reserved	Reserved	3 Bytes	Refer to the <i>SFF 8485 Specification for Serial GPIO (SGPIO) Bus</i> .

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4-15	Payload	Payload	48 bytes	The payload is 48 bytes in length. It is based on the SGPIO read/write command.

7.34 PCI_DIAG_EXECUTE Command

Description

This command is sent to the SPC 8x6G to initiate one of the PCIe Diagnostic Loopback tasks. The completion of this operation is reported using a PCIE_DIAG_EXECUTE Response as described in Section 8.36.

A PCIe diagnostic action is specified by a command, which is a combination of a command type (CMD_TYPE) and a command descriptor (CMD_DESC). See [Table 94](#) for a list of valid combinations of command types and command descriptors.

Usage Model and Limitation:

The intended use of this command is as follows:

1. Use only at power-on in both the manufacturing environment and fielded systems.
2. Execute when there is no other traffic occurring in the system before or while the PCIE Diagnostic tests are being run.
3. This command only supports a maximum of sixteen concurrent operations.
4. After the PCIe Diagnostic is complete, normal operation can begin without the need to reset.
5. Correct operation of this command or general operation of the controller is not guaranteed if this command is used when other operations are taking place in the SPC 8x6G.

Usage

Initiator and target.

Table 118 PCI_DIAG_EXECUTE Command Format

	Byte 3				Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x02F	
1	HTAG								

	Byte 3	Byte 2	Byte 1	Byte 0
2	Reserved		CMD_TYPE	CMD_DESC
3				
...			Reserved	
7				
8		READ_ADDR_LOWER		
9		READ_ADDR_UPPER		
10		WRITE_ADDR_LOWER		
11		WRITE_ADDR_UPPER		
12		LENGTH		
13		PATTERN		
14				
15		Reserved		

Table 119 PCI_DIAG_EXECUTE Command Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [15:13]	Command Type	CMD_TYPE	3 bits	000b: DIAG_OPRN_PERFORM: Performs a PCIe Diagnostic operation.
2 [12:8]	Command Descriptor	CMD_DESC	5 bits	00000b: WRITE_PATTERN 00001b: WRITE_BACK Also see Table 94 , which lists the command descriptor numeric assignment based on command type.
3-7	Reserved			
8 [31:0]	Lower read address bits	READ_ADDR_LOWER	32 bits	Lower 32 bits of the 64-bit host address to read data from (only used with WRITE_BACK command type).
9 [31:0]	Upper read address bits	READ_ADDR_UPPER	32 bits	Upper 32 bits of the 64-bit host address to read data from (only used with WRITE_BACK command type).
10 [31:0]	Lower write address bits	WRITE_ADDR_LOWER	32 bits	Lower 32 bits of the 64-bit host address to write data to.
11 [31:0]	Upper write address bits	WRITE_ADDR_UPPER	32 bits	Upper 32 bits of the 64-bit host address to write data to.
12 [31:0]	Data length	LENGTH	32 bits	Number of bytes of data to be read/written. Note that the current requirements limit the length to 16640 (520 * 32) bytes.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
13 [31:0]	Data pattern	PATTERN	32 bits	Data pattern to be used in conjunction with the WRITE_PATTERN command descriptor.

Table 120 PCIe Diagnostic Command Type and Command Description Valid Combination

Command Type (CMD_TYPE)	Command Descriptor (CMD_DESC)	Resulting Operation
DIAG_OPRN_PERFORM	WRITE_PATTERN	Write the specified pattern to host memory.
	WRITE_BACK	Read data from host memory and then write the data back.

Detailed Description of WRITE_PATTERN Command Type Operation

The WRITE_PATTERN command type supports transfer lengths between 8 and 16640 bytes (520 bytes * 32). The data in these transfers is 8-byte aligned.

The WRITE_PATTERN command type will write the 32-bit data pattern from the PATTERN field to host memory. The 64-bit host memory address is specified by the WRITE_ADDR_UPPER and WRITE_ADDR_LOWER fields. The number of bytes transferred is specified by the LENGTH field. The READ_ADDR_UPPER and READ_ADDR_LOWER fields are not used with this command type. Data will be transferred in 1024 byte packets.

Data written to the host is not checked by the SPC 8x6G. If the command fields are formatted correctly and the data is written successfully to the host, the response for this command will generate a SUCCESS status.

If the command fields are not formatted correctly (e.g., invalid length, invalid host address, etc.), the response for this command will generate a status of INVALID_COMMAND.

Detailed Description of WRITE_BACK Command Type Operation

The WRITE_BACK command type supports transfer lengths between 8 and 16640 bytes (520 bytes * 32). The data in these transfers is 8-byte aligned.

The WRITE_BACK command type will read data from the host at the address specified in the READ_ADDR_UPPER and READ_ADDR_LOWER fields. This data will then be written back to the host at the address specified in the WRITE_ADDR_UPPER and WRITE_ADDR_LOWER fields. Data will be transferred in 1024 byte packets.

Data read from and written to the host is not checked by the SPC 8x6G. If the command fields are formatted correctly and the data is written successfully to the host, the response for this command will generate a SUCCESS status.

If the command fields are not formatted correctly (e.g., invalid length, invalid host address, etc.), the response for this command will generate a status of INVALID_COMMAND.

8 Outbound Messages

The summary of outbound operation codes is provided in [Table 121](#).

Table 121 Outbound Operation Codes

Op Code (Hex)	Command Mnemonic	Usage Initiator/Target	IOMB Size (Bytes)	Section
0x0001	ECHO Response	I/T	64	8.1
0x0004	SAS_HW_EVENT Notification	I/T	64	8.2
0x0005	SSP_COMPLETION Response	I/T	Varies	8.3
0x0006	SMP_COMPLETION Response	I/T	Varies	8.4
0x0007	LOCAL_PHY_CONTROL Response	I/T	64	8.5
0x000A	DEVICE_REGISTRATION Response	I	64	8.6
0x000B	DREGISTER_DEVICE_HANDLE Response	I/T	64	8.7
0x000C	GET_DEVICE_HANDLE Response	I/T	Varies	8.8
0x000D	SATA_COMPLETION Response	I	Varies	8.9
0x000E	SATA_EVENT Notification	I	64	8.10
0x000F	SSP_EVENT Notification	I	64	8.11
0x0010	DEVICE_HANDLE_ARRIVED Notification	T	64	8.12
0x0012	SSP_REQUEST_RECEIVED Notification	T	Varies	8.13
0x0013	GET_DEVICE_INFO Response	I/T	64	8.14
0x0014	FW_FLASH_UPDATE Response	I/T	64	8.15
0x0016	GPIO Response	I/T	64	8.16
0x0017	GPIO_EVENT Notification	I/T	64	8.17
0x0018	GENERAL_EVENT Notification	I/T	64	8.18
0x001A	SSP_ABORT Response	I/T	64	8.19
0x001B	SATA_ABORT Response	I	64	8.20
0x001C	SAS_DIAG_MODE_START_END Response	I/T	64	8.21
0x001D	SAS_DIAG_EXECUTE Response	I/T	64	8.22
0x001E	GET_TIME_STAMP Response	I/T	64	8.23
0x001F	SAS_HW_EVENT_ACK Response	I/T	64	8.24
0x0020	PORT_CONTROL Response	I/T	64	8.25
0x0021	SKIP_ENTRIES_EVENT Notification	I/T	64	8.26
0x0022	SMP_ABORT Response	I/T	64	8.27
0x0023	GET_NVMD_DATA Response	I/T	64	8.28
0x0024	SET_NVMD_DATA Response	I/T	64	8.29
0x0025	DEVICE_HANDLE_REMOVED Notification	T	64	8.30
0x0026	SET_DEVICE_STATE Response	I/T	64	8.31
0x0027	GET_DEVICE_STATE Response	I/T	64	8.32

Op Code (Hex)	Command Mnemonic	Usage Initiator/Target	IOMB Size (Bytes)	Section
0x0028	SET_DEVICE_INFO Response	I/T	64	8.33
0x0029	SAS_RE_INITIALIZATION Response	I/T	64	8.34
0x082E	SGPIO Response	I	64	8.35
0x082F	PCIE_DIAG_EXECUTE Response	I/T	64	8.36

8.1 ECHO Response

Description

This response is sent by the SPC 8x6G controller as the response (echo) to the [ECHO Command](#) described in Section 7.1.

Usage

Initiator and target.

Response Format

Table 122 ECHO Response Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x001
1	HTAG							
2	PAYLOAD							
...								
15								

Table 123 ECHO Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as set in the ECHO Command .
[15:2]	Payload	PAYLOAD	56 Bytes	Host defined payload as set in the ECHO Command .

8.2 SAS_HW_EVENT Notification

Description

The notification is sent by the SPC 8x6G controller to alert the host that a hardware event that has occurred in the controller.

Local SAS/SATA port instantiation is reported through this hardware event with a link-up event type. See Section 3.1, “[SAS Port Instantiation, Port Context and PORT_ID](#)” for additional information.

[Table 125](#) contains detailed information about all the possible events.

The following events require host acknowledgement through the [SAS_HW_EVENT_ACK Command](#) (Section 7.24):

- IOP_HW_EVENT_PHY_DOWN (for most cases)
- IOP_HW_EVENT_BROADCAST_CHANGE
- IOP_HW_EVENT_PHY_ERROR_XX

Usage

Initiator and target.

Response Format

Table 124 SAS_HW Event Format

	Byte 3				Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x004							
1	Link Rate (LR)		STATUS		EVENT				PHYID	PORT_ID					
2	EVPARM														
3	Reserved						NPIP	PORT STATE							
4	SAS Identify Address Frame – First 28 Bytes (SASIDAF)														
...															
10															
11	Initial SATA FIS – 20 Bytes (INIFIS)														
...															
15															

Table 125 SAS_HW Event Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1 [31:28]	Link Rate	LR	4 bits	The PHY-negotiated Link Rate: 0001b: 1.5 Gbit/s. 0010b: 3 Gbit/s. 0100b: 6 Gbit/s. This field is only valid if EVENT is IOP_EVENT_SAS_PHY_UP, IOP_EVENT_SATA_PHY_UP, IOP_EVENT_PORT_RECOVER and IOP_EVENT_PORT_RESET_COMPLETE.
1 [27:24]	Additional status	STATUS	4 bits	Event specific additional status.
1 [23:8]	Event Type	EVENT	2 Bytes	0x0003: IOP_EVENT_PHY_STOP_STATUS. Event to notify the status of a PHY stop operation initiated by the PHY_STOP Command described in Section 7.3. The PHYID field contains the PHY identifier where this event is received. The STATUS field indicates the status of the operation: 0x00: SUCCESS 0x01: INVALID_PHY 0x04: PHY_NOT_ENABLED. An attempt to stop a PHY not yet started 0x03: OTHER_FAILURE The EVPARM field indicates the HTAG passed in PHY_STOP Command . There is no subsequent IOP_EVENT_PHY_DOWN notification to the host at the completion of successful PHY stop operation.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>0x0004: IOP_EVENT_SAS_PHY_UP. The local PHY has achieved PHY READY state and a SAS identify address frame is received from a SAS device. If this is the first PHY that received this event for the local SAS port (either narrow or wide port), the PORT_ID field represents a new controller SAS port instantiation. The PHYID field contains the PHY identifier where this event is received. The NPIP represents the number of PHYs belonging to the PORT_ID that are in ready (PHY up) state. The PS is set to PORT_VALID indicating the port specified by PORT_ID is valid. The SASIDAF field contains the Identify address frame of the directly attached SAS device.</p> <p>If the PORT_ID reported is a new port, the host should use the value in SASIDAF field to do device registration through the REGISTER_DEVICE Command described in Section 7.15. Only one device registration is needed per port to register the directly-attached device.</p> <p>If the PORT_ID reported is an existing port, the host should not do a new device registration. This is the case during PHY down handle (Section 11.5.1, “PHY Down Handler (External Trigger)”) and local link/hard reset PHY control (Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)”).</p> <p>Other unused fields are invalid.</p> <p>0x0005: IOP_EVENT_SATA_PHY_UP. The local PHY has achieved PHY READY state and an initial FIS is received from a SATA device. The PORT_ID field represents a new controller SATA port instantiation. The PHYID field contains the PHY identifier where this event is received. The NPIP represents the number of PHYs belonging to the PORT_ID that are in ready (PHY up) state, which is always 1 for a directly-attached SATA drive. The PS is set to PORT_VALID indicating the port specified by PORT_ID is valid. The INIFIS field contains the initial FIS of the directly attached SATA device.</p> <p>If the PORT_ID reported is a new port, the host should use the value in the PHYID field to register the directly attached SATA drive through the REGISTER_DEVICE Command described in Section 7.15.</p> <p>If the PORT_ID reported is an existing port, the host should not do a new device registration. This is the case during PHY down handle (Section 11.5.1, “PHY Down Handler (External Trigger)”) and local link/hard reset PHY control (Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)”).</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>Other unused fields are invalid.</p> <p>0x0006: IOP_EVENT_SATA_SPINUP_HOLD. Indicates a SATA device directly attached to a local PHY is in the spin-up hold state. The IOP_EVENT_SATA_PHY_UP event is not received for this PHY until spin-up hold is released via the LINK RESET operation in the LOCAL_PHY_CONTROL Command described in Section 7.18. The PHYID field contains the PHY Identifier where this event is received. Other unused fields are invalid.</p> <p>0x0007: IOP_EVENT_PHY_DOWN. The PHY is in the link down state. The PHYID field contains the PHY Identifier where this event is received. The NPIP represents the number of PHYs belonging to PORT_ID that are still in ready (PHY up) state.</p> <p>PS indicates the Port State of the port specified by the PORT_ID.</p> <p>If this PHY is not the last PHY in the port, the PS will be set to PORT_VALID and NPIP to a non-zero value.</p> <p>For an external triggered PHY down (Section 11.5.1, “PHY Down Handler”), if this PHY is the last PHY in the port:</p> <ul style="list-style-type: none"> • If Port Recovery Time is set to non-zero, the PS will be set to PORT_LOSTCOMM and NPIP to zero. • If Port Recovery Time is set to zero, the PS will be set to PORT_INVALID and NPIP to zero. <p>For a host-initiated local PHY control link/hard reset (Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)”), if this PHY is the last PHY in the port:</p> <ul style="list-style-type: none"> • The PS will be set to PORT_IN_RESET and NPIP to zero. <p>Other unused fields are invalid.</p> <p>This event may require a host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24) for releasing the PORT_ID when the last PHY in a port is down (port state is OSSA_PORT_INVALID). See Section 11.5.1, “PHY Down Handler” for a description of the PHY down event condition</p> <p>This event is also reported as the result of LINK RESET and HARD_RESET operation in the LOCAL_PHY_CONTROL Command. (See Section 7.18.)</p> <p>Sending the PHY down acknowledgement through a SAS_HW_EVENT_ACK Command (see Section</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>7.24) in some cases is done after subsequent reception of other event such as IOP_EVENT_PORT_RECOVERY_TIMER_TMO or IOP_EVENT_PORT_RESET_TIMER_TMO when the PS is becoming PORT_INVALID.</p> <p>This event is not reported as the subsequent notification to IOP_EVENT_PHY_STOP_STATUS.</p> <p>0x0009: IOP_EVENT_BROADCAST_CHANGE. The corresponding PHY has received a SAS Broadcast change primitive as a notification of configuration change. The host needs to re-initiate discovery from the associated local port. The PHYID field contains the PHY Identifier where this event is received. Other unused fields are invalid.</p> <p>This event required host acknowledgement through a SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, “BROADCAST CHANGE Handler”.</p> <p>0x000B: IOP_EVENT_BROADCAST_SES. The corresponding PHY has received a SAS Broadcast SES primitive as a notification of a configuration change. The host reads the SES information from the associated local port. The PHYID field contains the PHY identifier where this event is received. Other unused fields are invalid.</p> <p>0x000C: IOP_EVENT_PHY_ERR_INBOUND_CRC. The PHY has detected an inbound frame with a CRC error. The PHYID field contains the PHY Identifier where this event is received. The EVPARM field indicates the current inbound CRC error count.</p> <p>This event requires host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, “BROADCAST CHANGE Handler”.</p> <p>0x000D: IOP_EVENT_HARD_RESET_RECEIVED. The PHY has received a hard reset. See Section 11.5.5, “Hard Reset Received Handler” for handling this event. The PHYID field contains the PHY identifier where this event is received. Other unused fields are invalid.</p> <p>0x000F: IOP_EVENT_ID_FRAME_TIMEOUT. The IDENTIFY frame timeout is detected on the corresponding PHY. The PHYID field contains the PHY identifier where this event is received. Other unused fields are invalid.</p> <p>0x0010: IOP_EVENT_BROADCAST_EXP. The corresponding PHY has received a SAS Broadcast (Expander) primitive as PHY event notification from the expander. The host reads the expander information from the associated local port. The PHYID field contains the PHY identifier where this event is received. Other unused fields are invalid.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>0x0011: IOP_EVENT_PHY_START_STATUS. Event to notify the status of a PHY start operation initiated by the PHY_START Command (see Section 7.2). The PHYID field contains the PHY identifier where the event is received. The STATUS field indicates the status of the operation:</p> <ul style="list-style-type: none"> 0x00: SUCCESS 0x01: INVALID_PHY 0x02: PHY_NOT_DISABLED. An attempt to start a PHY that is already started. 0x03: OTHER_FAILURE <p>The EVPARM field indicates the HTAG passed in PHY_START Command.</p> <p>0x0012: IOP_EVENT_PHY_ERR_INVALID_DWORD. The PHY has detected a invalid DWord link error. The PHYID field contains the PHY identifier where this event is received. The EVPARM field indicates the current invalid DWord error count.</p> <p>This event requires host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, "BROADCAST CHANGE Handler".</p> <p>0x0013: IOP_EVENT_PHY_ERR_DISPARITY_ERROR. The PHY has detected a disparity link error. The PHYID field contains the PHY identifier where this event is received. The EVPARM field indicates the current disparity error count.</p> <p>This event requires host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, "BROADCAST CHANGE Handler".</p> <p>0x0014: IOP_EVENT_PHY_ERR_CODE_VIOLATION. The PHY has detected a code violation error. The PHYID field contains the PHY identifier where this event is received. The EVPARM field indicates the current code violation error count.</p> <p>This event requires host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, "BROADCAST CHANGE Handler".</p> <p>0x0015: IOP_EVENT_PHY_ERR_LOSS_OF_DWORD_SYNCH. The PHY has detected a loss of DWord synchronization errors. The PHYID field contains the PHY identifier where this event is received. The EVPARM field indicates the current loss of the DWord synchronization count.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>This event requires host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, “BROADCAST CHANGE Handler”.</p> <p>0x0016: IOP_EVENT_PHY_ERR_PHY_RESET_FAILED. The PHY has detected a PHY reset fail error. The PHYID field contains the PHY identifier where this event is received. The EVPARM field indicates the current PHY reset fail count.</p> <p>This event requires host acknowledgement through the SAS_HW_EVENT_ACK Command (see Section 7.24). Also see Section 11.5.3, “BROADCAST CHANGE Handler”.</p> <p>0x0017: IOP_EVENT_PORT_RECOVERY_TIMER_TMO. This event will only be reported if the Port Recovery Time is set to a non-zero value. The port whose last PHY was previously reported in link down state (with event IOP_EVENT_PHY_DOWN) was not able to recover within the Port Recovery Time.</p> <p>The PHYID field contains the PHY Identifier where the PHY down event was previously received in the event of an IOP_EVENT_PHY_DOWN.</p> <p>PS indicates the Port State of the port specified by the PORT_ID. The PS field will be set to PORT_INVALID and the NPIP field to zero. Other unused fields are invalid.</p> <p>See Section 11.5.1, “PHY Down Handler” for the detail information.</p> <p>0x0018: IOP_EVENT_PORT_RECOVER. This event will only be reported if the Port Recovery Time is set to a non-zero value. The port whose last PHY was previously reported in the link down state (with event IOP_EVENT_PHY_DOWN) has recovered within the Port Recovery Time.</p> <p>The PHYID field contains the PHY Identifier where this event is received. The PHY Identifier could either be the same PHY as previously reported in the IOP_EVENT_PHY_DOWN or a different PHY belonging to the same port.</p> <p>Note: The remote PHY Identifier is not checked so technically two remote PHYs belonging to the same wide port could be swapped on the same port be reported in the IOP_EVENT_PORT_RECOVER event.</p> <p>PS indicates the Port State of the port specified by the PORT_ID. The PS field will be set to PORT_VALID and the NPIP field to one. The SASIDAF field contains the Identify address frame of the directly-attached SAS device. The LR field will</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>have the value for the new PHY-negotiated Link Rate. Other unused fields are invalid.</p> <p>See Section 11.5.1, “PHY Down Handler” for details.</p> <p>0x0019: IOP_EVENT_PORT_RESET_TIMER_TMO. This event is reported when the Port Reset Time has expired and no PHY up is detected in any of the PHY in that port. See Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for the description of the Port Reset Time.</p> <p>0x0020: IOP_EVENT_PORT_RESET_COMPLETE. This event is reported when the host initiated local PHY control link/hard reset has completed successfully with at least one PHY is up in the port. PS is set to PORT_VALID and I/O can be resumed. See Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for the description of this event. The SASIDAF field contains the Identify address frame of the directly-attached SAS device. The LR field will have the value for the new PHY-negotiated Link Rate. Other unused fields are invalid.</p> <p>0x0021: IOP_EVENT_BROADCAST_ASYNC_EVENT. The corresponding PHY has received a SAS Broadcast Asynchronous Event primitive as a notification of a SSP target port event (e.g., a hard reset that causes one or more unit attention conditions to be established for one or more logical units accessible through the SSP target port). The PHYID field contains the PHY identifier where this event is received. The PS field will be set to PORT_VALID and the NPIP field to non-zero. Other unused fields are invalid.</p> <p>0x0022: IOP_EVENT_IT_NEXUS_LOSS. [Optional] An I/O sent to a device experiences IT nexus loss. This is an optional event sent only when bit #1 of the MPI Main Configuration Table (DW#0x13) is set. EVPARAM field is the device ID (32 bits) where the event is received.</p> <p>The STATUS field is the status of the operation:</p> <ul style="list-style-type: none"> 0x00: SAME. Device did not receive IT Nexus Loss timeout. 0x01: NON_OPERATIONAL. Device was already set to non-operational. 0x02: UNUSED. Device was deregistered. 0x03: REUSED. Device was set by a TM command to DS_IN_RECOVER or the same device ID (lower 16 bits) was deregistered and reassigned. <p>0x0023: IOP_EVENT_PHY_LOCKUP_DATAOUT [Optional] The SPC resolves a lockup by resetting</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>the hardware at the PHY from which the stuck I/O is sent.</p> <p>The stuck I/O will get an SSP_EVENT(IO_XFER_ERR_PHY_NOT_READY) after the SPC brings down the PHY. The host will experience a hotplug sequence lasting only for a few milliseconds (SNW1/SNW2) or a few tens of milliseconds (SNW3)</p> <p>0x0024: IOP_EVENT_PHY_DOC_ABORT_TMO. [Optional] The SPC experiences a timeout waiting for a write command aborted in the Data phase. The timeout is specified in the IO Abort Delay field of the MPI Configuration Table. See Table 38. This is an optional event sent only when bit #3 of the MPI Configuration Table DW #13 is set. See section 3.24.4. PHYID is the PHY at which the lockup happens. PORT_ID, NPIP and PS are set accordingly. EVPARM contains I/O's HTAG.</p>
1 [7:4]	PHY Identifier	PHYID	4 bits	Zero-based PHY identifier.
1 [3:0]	Port Identifier	PORT_ID	4 bits	<p>The ID for the SPC 8x6G controller's local SAS/SATA port context.</p> <p>See Section 3.1, "SAS Port Instantiation, Port Context and PORT_ID" for more details about port instantiation.</p> <p>This port context may either be a new port context for the newly instantiated local port or an existing port context that has been reported previously with the SAS_HW_EVENT Notification.</p> <p>The host should check for a newly instantiated port (and PORT ID) by comparing this value with the previously reported port context.</p>
2	Event Parameters	EVPARM	4 Bytes	Event specific parameters status. Zero if not used.
3 [7:4]	Number of PHYs in Port	NPIP	4 bits	Number of PHYs in ready state (PHY up) for the corresponding PORT_ID. A value of 1 indicates narrow-port, a value larger than 1 indicates wide-port, and a value of zero indicates that all PHYs belonging to the PORT_ID are down.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
3 [3:0]	Port State	PS	4 bits	<p>0000b: PORT_NOT_ESTABLISHED: The port state is not established yet, the PORT_ID field is not valid.</p> <p>0001b: PORT_VALID: The port specified by PORT_ID is valid.</p> <p>0110b: PORT_LOSTCOMM: The port is temporarily unavailable due to a temporary PHY down of the last PHY in the port. This state only exists if the Port Recovery Time parameter is set to a non-zero value and externally triggered PHY down is detected.</p> <p>0100b: PORT_IN_RESET: The port is temporarily unavailable due to a temporary PHY down of the last PHY in the port. This state only exists when the host initiated the local PHY control of link/hard reset. See Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for the description of the PORT_IN_RESET state.</p> <p>1000b: PORT_INVALID: The port specified by PORT_ID is becoming invalid, i.e. all PHYs are down.</p> <p>Others: Reserved.</p>
[10:4]	SAS Identify Address Frame	SASIDAF	28 Bytes	The first 28 bytes of the SAS_IDENTIFY address frame (excluding the CRC). Big Endian format. See Section 2.1.3.1, “SAS Payload Endianness” for details about SAS payload endianness.
[15:11]	Initial SATA FIS	INIFIS	20 Bytes	Initial FIS of the directly-attached SATA device. Little Endian format. See Section 2.1.3.2, “SATA Payload Endianness” for details about SATA payload endianness.

8.3 SSP_COMPLETION Response

Description

This response is sent by the SPC 8x6G controller to indicate a completion of an SSP request previously issued to the controller through a:

- [SSP_INI_IO_START Command](#) (Section 7.4)
- [SSP_INI_EXT_IO_START Command](#) (Section 7.6)
- [SSP_INI_TM_START Command](#) (Section 7.5)
- [SSP_TGT_IO_START Command](#) (Section 7.8)
- [SSP_TGT_RESPONSE_START Command](#) (Section 7.9)
- [SSP_ABORT Command](#) (Section 7.10)

The host can safely free the resources allocated for the associated operation listed above.

Usage

Initiator and target.

Response Format

Table 126 SSP_COMPLETION Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=n		Reserved	OBID	CAT=0x02		
1	HTAG									
2	STATUS									
3	PARAM									
4	Reserved			AGR	RESC_P AD	RESC_V	SSP TAG			
5	SSP_RESPONSE IU – If exists (SSPRSP)									
...										
N										
N+1	RESIDUAL_COUNT (RESC) – if existing (RESC_V is set)									

Table 127 SSP_COMPLETION Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
0 [3:0]	Buffer Count	BC	5 bits	<p>Buffer count.</p> <p>The SSP_COMPLETION Response can use more than one buffer/message entry depending on whether there is an SSP_RESPONSE_INFORMATION_UNIT reported.</p>
1	Host Tag	HTAG	4 Bytes	<p>Tag or context as specified in the SSP_XX_XX_START Command.</p>
2	SSP Completion Status	STATUS	4 Bytes	<p>Status of the operation.</p> <p>0x00000000: IO_COMPLETED. An I/O request completed successfully. If the PARAM field is zero, the I/O command is completed successfully. (For initiator mode, this implies the SCSI status is GOOD_STATUS). The host does not need to do any additional action. The SSP RESP IU field is not valid. If the PARAM field is non-zero (initiator only), the SSP_RESP_IU is present, and the PARAM field contains the number of bytes of SSP_RESP_IU transferred to the host. This indicates that an I/O request has completed with a SCSI status other than GOOD_STATUS. It could also indicate a task management command completion. The SSP RESP IU field contains at least the first 24 bytes of SSP_RESPONSE IU. The host should read SENSE DATA LENGTH (bytes [19:16] of SSP response IU) and RESPONSE DATA LENGTH (bytes [23:20] of SSP response IU) to know the actual total SSP response length.</p> <p>0x00000001: IO_ABORTED. An I/O aborted successfully. This response is a result of a previous SSP_ABORT Command (Section 7.10). The PARAM field contains additional information:</p> <ul style="list-style-type: none"> 0: Aborted by an abort command. 1: Aborted by an a TM command. Other values: Reserved. <p>0x00000003: IO_UNDERFLOW. An I/O had a data underrun. The PARAM field indicates that the residual bytes were not transferred.</p> <p>0x00000007: IO_NO_DEVICE. An I/O request was sent to a non-existent device (invalid DEVICE_ID). The PARAM field is not used.</p> <p>0x0000000E: IO_XFER_ERROR_BREAK. In target mode, an I/O aborted due to a connection error caused by a BREAK. The PARAM field is not used.</p> <p>0x0000000F: IO_XFER_ERR_PHY_NOT_READY. An I/O aborted due to a PHY_NOT_READY during a</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>connection in the command phase. The PARAM field is not used.</p> <p>0x00000010: IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED. An I/O request failed due to an open connection error during the command phase. The host needs to do a rediscovery. The PARAM field is not used.</p> <p>0x00000011: IO_OPEN_CNX_ERROR_ZONE_VIOLATION. An I/O request failed due to an open connection error during the command phase. The host needs to do a rediscovery. The PARAM field is not used.</p> <p>0x00000012: IO_OPEN_CNX_ERR_BREAK. An I/O request failed due to an open connection error because a BREAK was received. The PARAM field is not used.</p> <p>0x00000013: IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. An I/O request failed due to an open connection error during the command phase. The SPC 8x6G will automatically set the device state to DS_NON_OPERATIONAL. The host is required to complete the recovery procedure as described in Section 11.5.6, “Device States Related Error Handler” and Section 11.5.6.3, “Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices”. The PARAM field is not used.</p> <p>0x00000014: IO_OPEN_CNX_ERROR_BAD_DESTINATION. An I/O request failed due to an open connection error during the command phase. The PARAM field is not used.</p> <p>0x00000015: IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED. An I/O request failed due to an open connection error during the command phase. The PARAM field is not used.</p> <p>0x00000017: IO_OPEN_CNX_ERROR_WRONG_DESTINATION. An I/O request failed due to an open connection error during the command phase. The PARAM field is not used.</p> <p>0x00000019: IO_XFER_ERROR_NAK RECEIVED. An I/O aborted, due to a transfer error with a NAK received for the command frame. The PARAM field is not used.</p> <p>0x0000001A: IO_XFER_ERROR_ACK_NAK_TIMEOUT. (Target mode). The SPC 8x6G detects an</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>ACK/NAK timeout during the data phase, response phase or the transfer ready phase. The HTAG field represents the TAG originally passed in the SSP_TGT_IO_START Command.</p> <p>0x0000001D: IO_XFER_ERROR_DMA. An I/O aborted due to an error previously detected in the data transfer phase (e.g. data overflow when SSP_EVENT of IO_OVERFLOW was previously posted for this IO). The PARAM field is not used</p> <p>0x00000024: IO_XFER_OPEN_RETRY_TIMEOUT. An I/O aborted due to a open retry TIMEOUT when trying to send the command frame. The PARAM field is not used.</p> <p>0x00000026: IO_XFER_ERROR_UNEXPECTED_PHASE: In target only, the SPC 8x6G receives data when it is not expecting any.</p> <p>0x00000034: IO_XFER_ERROR_OFFSET_MISMATCH. In target mode, the SPC 8x6G detects during a SCSI WRITE operation that the DATA_OFFSET field in the SSP frame header is out of sequence.</p> <p>0x00000038: IO_PORT_IN_RESET. An I/O aborted due to port in reset state. See Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for port reset state. The PARAM field is not used.</p> <p>0x00000039: IO_DS_NON_OPERATIONAL. An I/O aborted due to the device state in the DS_NON_OPERATIONAL state. See Table 8 in Section 3.2, “Device Handle and DEVICE_ID” for the definition of the device states</p> <p>0x0000003A: IO_DS_IN_RECOVERY. An I/O aborted due to the device state in the DS_IN_RECOVERY state. See Table 8 in Section 3.2, “Device Handle and DEVICE_ID” for the definition of the device states.</p> <p>0x0000003B: IO_TM_TAG_NOT_FOUND. An ABORT_TASK task management request sent through the SSP_INI_TM_START Command, as described in Section 7.5, contains an invalid HTAG-A.</p> <p>0x0000003D: IO_SSP_EXT_IU_ZERO_LEN_ERROR. An I/O aborted due to a zero length specified in the SSPIUL field when sending the SSP_INI_EXT_IO_START Command described in Section 7.6. The PARAM field is not used.</p> <p>0x0000003F: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY. An I/O request failed due to an open connection</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>error. The SPC 8x6G cannot allocate the hardware resource when it tries to open a connection to the target. Host needs to retry that I/O request.</p> <p>0x00000043: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT. An I/O request failed due to an open connection error. The SPC 8x6G drains the I/O to prevent a lockup. See Section 3.24.1.</p>
3	SSP Completion Parameters	PARAM	4 Bytes	Additional information. The definition of this field depends on the STATUS field.
4 [15:0]	SSP Initiator TAG	SSP TAG	2 Bytes	<p>In initiator mode, it is the SSP (Initiator) TAG defined as the TAG field in bytes 16 and 17 of the SSP frame. In target mode, it is Target Port Transfer Tag (TPTT) Bytes 18 and 19 of the SSP frame.</p> <p>This field is used for debugging purposes, i.e. matching SSP completion with the SAS trace in the SAS analyzer.</p>
4 [16]	Residual Count Valid	RESC_V	1 bit	<p>This bit indicates that the RESIDUAL_COUNT field is valid.</p> <p>This field is valid in initiator mode and when the STATUS is IO_COMPLETED.</p>
4 [18:17]	Residual Count Pad Count	RESC_PAD	2 bits	When RESC_V is 1, this field contains the number of bytes padded to the end of the SSP_RESPONSE_IU field to make the SSP_RESPONSE_IU 4-byte aligned. This value, when added to the value passed in PARAM field, can be used as an offset to retrieve the RESIDUAL_COUNT value.
4 [19]	Auto Good Response Status	AGR_S	1 bits	<p>This field is applicable to Target Mode only.</p> <p>When the target host application issues a SSP_TGT_IO_START IOMB with the AGR bit set and there is an error in operation, this field indicates the following:</p> <p>0b : the failure occurred in the Data Phase 1b : the failure occurred in the Response Phase</p> <p>If this bit is set and the SPC 8x6G reports the status of IO_XFER_OPEN_RETRY_TIMEOUT, the error occurred during the connection management of the RESPONSE phase; therefore, the host may choose to reissue the response using SSP_TGT_RESPONSE_START</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
5:N	SSP_RESPONSE IU	SSPRSP	(N-4) *4 Bytes	The SSP Response Information Unit. For initiator mode only. Only available when the STATUS field is IO_COMPLETED and the PARAM field is non-zero. The host should read the DATAPRES field to determine if either RESPONSE_DATA or SENSE_DATA is present and process the response accordingly.
N+1	RESIDUAL-COUNT	RESC	4 Bytes	This field is only valid if the RESC_V field is set. It indicates that an I/O (initiator) with a data underrun during the data phase is reported with a non-zero status. This field contains the residual bytes that are not transferred during the I/O operation.

8.4 SMP_COMPLETION Response

Description

This response is sent by the SPC 8x6G controller to indicate the completion of an SMP request previously issued to the controller by the [SMP_REQUEST Command](#) (Section 7.13).

In target mode, this response is sent after the completion of a data transfer operation.

In initiator mode, this response carries the SMP_RESPONSE frame associated with the SMP_REQUEST initiated by the [SMP_REQUEST Command](#).

Usage

Initiator and target.

Response Format

Table 128 SMP_COMPLETION Response Format

	Byte 3				Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=n	Reserved	OBID	CAT=0x02	OPC=0x006			
1	HTAG										
2	STATUS										
3	PARAM										
4	SMP_RESPONSE Frame – Initiator only (SMPRESP)										
...											
N											

Table 129 SMP_COMPLETION Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
0 [3:0]	Buffer Count	BC	5 bits	Buffer count. The SMP_COMPLETION response can use more than one buffer/message entry depending on whether there is an SMP_RESPONSE reported.
1	Host Tag	HTAG	4 Bytes	Tag or context as set in the SMP_REQUEST Command (Section 7.13).
2	SMP Completion Status	STATUS	4 Bytes	Status of the operation: 0x00000000: IO_COMPLETED. An SMP request completed successfully. For an initiator, when the IR flag in the SMP_REQUEST Command is set to 0b (direct mode), the PARAM field describes the length of the SMP response in the Response IOMB, and when the IR flag in the SMP_REQUEST Command is set to 1b (indirect mode), the PARAM field describes the length of the SMP response that has been DMA'd into the response buffer. 0x00000001: IO_ABORTED. An SMP request is aborted successfully. This response is a result of a previous SMP_ABORT Command (Section 7.14). The PARAM field contains additional information: 0: Aborted by an abort command. Other values: Reserved. 0x00000002: IO_OVERFLOW. The SPC 8x6G detects an overflow during the indirect mode operation. SMP operation will be completed normally. The HTAG field represents the TAG originally passed in the SMP_REQUEST Command (Section 7.13).

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>0x00000007: IO_NO_DEVICE. An SMP request was sent to a non-existent device (invalid DEVICE_ID).</p> <p>0x0000000D: IO_ERROR_HW_TIMEOUT. An SMP request does not receive any SMP responses within 2 ms.</p> <p>0x0000000E: IO_XFER_ERR_BREAK. An SMP request failed because the SPC 8x6G has transmitted or received a BREAK during an SMP connection. The PARAM field is not used.</p> <p>0x0000000F: IO_XFER_ERR_PHY_NOT_READY. An SMP request failed due to a PHY_NOT_READY during connection. The PARAM field is not used.</p> <p>0x00000010: IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED. An SMP request failed due to an open connection error. The host must do a rediscovery. The PARAM field is not used.</p> <p>0x00000011: IO_OPEN_CNX_ERROR_ZONE_VIOLATION. An SMP request failed due to an open connection error. The host must do a rediscovery. The PARAM field is not used.</p> <p>0x00000012: IO_OPEN_CNX_ERROR_BREAK. An SMP request failed due to an open connection error because a BREAK was received. The PARAM field is not used.</p> <p>0x00000013: IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. An SMP request failed due to an open connection error. The SPC 8x6G will automatically set the device state to DS_NON_OPERATIONAL. The host is required to complete the recovery procedure as described in Section 11.5.6, “Device States Related Error Handler” and Section 11.5.6.3, “Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices”. The PARAM field is not used.</p> <p>0x00000014: IO_OPEN_CNX_ERROR_BAD_DESTINATION. An SMP request failed due to an open connection error. The PARAM field is not used.</p> <p>0x00000015: IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED. An SMP request failed due to an open connection error. The PARAM field is not used.</p> <p>0x00000017: IO_OPEN_CNX_ERROR_WRONG_DESTINATION. An SMP request failed due to an open</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>connection error. The PARAM field is not used.</p> <p>0x0000001C: IO_XFER_ERROR_RX_FRAME. An SMP request failed due to the SMP response frame being received with any of the following errors: SMP response frame length = 0, SMP response frame is too long.</p> <p>0x00000024: IO_XFER_OPEN_RETRY_TIMEOUT: An SMP request failed due to an open retry TIMEOUT. The PARAM field is not used.</p> <p>0x00000037: IO_ERROR_INTERNAL_SMP_RESOURCE. An SMP request failed due to the lack of an internal SMP resource. Try sending the request later. The PARAM field is not used.</p> <p>0x00000038: IO_PORT_IN_RESET. An SMP request failed due to the port being in reset state. See Section 11.5.2, "Local PHY Control Link/Hard Reset Handler (Host Initiated)" for a description of the port reset state. The PARAM field is not used.</p> <p>0x00000039: IO_DS_NON_OPERATIONAL. An SMP request failed due to being in the DS_NON_OPERATIONAL device state. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states.</p> <p>0x0000003A: IO_DS_IN_RECOVERY. An SMP request failed due to being in the DS_IN_RECOVERY device state. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states.</p> <p>0x0000003F: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY. An SMP request failed due to an open connection error. The SPC 8x6G cannot allocate the hardware resource when it tries to open an SMP connection to the target. After failing to allocate the resource for 100~200 µs, the SPC 8x6G terminates the request with this error code.</p> <p>0x00000042: IO_INVALID_LENGTH. An SMP request failed due the SMP_REQUEST Command specifying the wrong value for the Indirect SMP payload length and/or Indirect SMP_RESPONSE frame length.</p> <p>0x00000043: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT. An I/O request failed due to an open connection error. The SPC 8x6G drains the I/O to prevent a lockup. See Section 3.24.1.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
3	SMP Completion Parameters	PARAM	4 Bytes	The number of bytes of the SMP response associated with the SMP request.
4:N	SMP_RESPONSE Frame	SMPRSP	(N-3) * 4 Bytes	<p>The SMP Response frame. For initiator mode only, and only available when all of the following conditions are met:</p> <ul style="list-style-type: none"> • STATUS is IO_COMPLETED. • PARAM is non-zero. • IR flag in SMP_REQUEST Command (Section 7.13) is set to 0b. <p>Big Endian format. See Section 2.1.3.1, “SAS Payload Endianness” for details about the payload endianness.</p>

8.5 LOCAL_PHY_CONTROL Response

Description

This response is sent by the SPC 8x6G controller as the response to the previous [LOCAL_PHY_CONTROL Command](#), which is described in Section 7.18.

Usage

Initiator and target.

Response Format

Table 130 LOCAL_PHY_CONTROL Response Format

	Byte 3				Byte 2		Byte 1		Byte 0								
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x007									
1	HTAG																
2	Reserved				PHYOP		Reserved	PHYID									
3	STATUS																
4	Reserved																
...																	
15																	

Table 131 LOCAL_PHY_CONTROL Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the LOCAL_PHY_CONTROL Command .
2 [3:0]	PHY Identifier	PHYID	4 bits	PHY Identifier previously passed in the LOCAL_PHY_CONTROL Command .
2 [15:8]	PHY Operation	PHYOP	1 Byte	PHY Operation previously passed in the LOCAL_PHY_CONTROL Command .
3	Status	STATUS	4 Bytes	Status of the operation: 0x00: SUCCESS. The LOCAL_PHY_CONTROL operation completed successfully. 0x03: FAILURE. The LOCAL_PHY_CONTROL operation failed.

8.6 DEVICE_REGISTRATION Response

Description

This response is sent by the SPC 8x6G controller to notify the completion for the host initiated new device registration previously initiated by the [REGISTER_DEVICE Command](#) described in Section [7.15](#).

A successful registration status returns the SPC 8x6G-assigned DEVICE_ID for the corresponding device.

If the host sets the H bit in the [REGISTER_DEVICE Command](#) for the host-assigned UPPER DEVICE_ID, the SPC 8x6G will retain this value in the upper 16 bits of the DEVICE_ID. If the host does not set the H bit in the [REGISTER_DEVICE Command](#) for the host-assigned UPPER DEVICE_ID, the SPC 8x6G will only assign the lower 16 bits of the DEVICE_ID.

The host may use the lower 16 bits of the DEVICE_ID for host indexing. The lower 16 bits are always an SPC 8x6G-assigned value. The full 32-bit DEVICE_ID is always required when exchanging DEVICE_ID fields between the host and the SPC 8x6G.

Usage

Initiator.

Response Format

Table 132 DEVICE_REGISTRATION Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02	OPC=0x00A	
1	HTAG									
2	STATUS									
3	DEVICE_ID									
4	Reserved									
...										
15										

Table 133 DEVICE_REGISTRATION Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the REGISTER_DEVICE Command .

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	Device Registration Status	STATUS	4 Bytes	<p>0x00000000: SUCCESS. The device registration completed successfully. The parameter DEVICE_ID specifies the ID for the newly assigned device handle.</p> <p>0x00000001: FAILURE OUT OF RESOURCE. The device registration failed because the SPC 8x6G is running out of device handle resources. The parameter DEVICE_ID is not used.</p> <p>0x00000002: FAILURE DEVICE ALREADY REGISTERED. The device registration failed because the SPC 8x6G detected an existing device handle with a similar SAS address. The parameter DEVICE_ID contains the existing DEVICE_ID assigned to the SAS device.</p> <p>0x00000003: FAILURE INVALID PHY ID. Only for directly -attached SATA registration. The device registration failed because the SPC 8x6G detected an invalid (out-of-range) PHY ID.</p> <p>0x00000004: FAILURE PHY ID ALREADY REGISTERED . Only for directly -attached SATA registration. The device registration failed because the SPC 8x6G detected an already -registered PHY ID for a directly- attached SATA drive.</p> <p>0x00000005: FAILURE PORT ID OUT OF RANGE. PORT_ID specified in the REGISTER_DEVICE Command is out-of-range (0-7).</p> <p>0x00000006: FAILURE PORT NOT VALID STATE. The PORT_ID specified in the REGISTER_DEVICE Command is not in PORT_VALID state.</p> <p>0x00000007: FAILURE DEVICE TYPE NOT VALID. The device type, specified in the 'S' field in the REGISTER_DEVICE Command is not valid.</p>
3	Device Identifier	DEVICE_ID	4 Bytes	The newly assigned device identifier. See Section 3.2 for a detailed description of DEVICE_ID.

8.7 DREGISTER_DEVICE_HANDLE Response

Description

This response is sent by the SPC 8x6G controller to notify the completion for removing the device handle previously initiated by the [DREGISTER_DEVICE_HANDLE Command](#) described in Section [7.11](#).

Usage

Initiator and target.

Response Format

Table 134 DREGISTER_DEVICE_HANDLE Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02 OPC=0x00b		
1	TAG									
2	STATUS									
3	Reserved									
...										
15										

Table 135 DREGISTER_DEVICE_HANDLE Response Fields

DWord	Field	Length	Description
1	TAG	4 Bytes	Tag or context for this operation.
2	STATUS	4 Bytes	Status of operation: SUCCESS (0x00000000) : Successfully removed the SPC 8x6G resource associated with this device handle (DEVICE_ID). INVALID HANDLE (0x00000002) : The device handle associated with DEVICE_ID does not exist.
[15:3]	Reserved	52 Bytes	Reserved.

8.8 GET_DEVICE_HANDLE Response

Description

This message is sent by the SPC 8x6G controller to return the device handles (DEVICE_IDs) as the response to [GET_DEVICE_HANDLE Command](#) described in Section [7.12](#).

Usage

Initiator and target.

Response Format

Table 136 GET_DEVICE_HANDLE Response Format

	Byte 3			Byte 2		Byte 1		Byte 0			
0	V	H	R	BC=n		Reserved	OBID	CAT=0x02 OPC=0x00c			
1	HTAG										
2	Reserved			DEVICE_IDC			Reserved	PORT_ID			
3	DEVICE_ID (0)DEVICE_ID										
...											
N+2	DEVICE_ID (N-1)DEVICE_ID										

Table 137 GET_DEVICE_HANDLE Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [23:8]	Device Identifier count	DEVICE_IDC	2 Bytes	Number of device handles returned. Since the maximum size of the (concatenated) outbound IOMB for the response is limited to 1024 bytes, the maximum number of device handles returned is (256-3) entries.
2 [3:0]	Port Identifier	PORT_ID	4 bits	SPC 8x6G local SAS/SATA port context Identifier.
3..(N+2)	Device Identifier	DEVICE_ID	N*4 Bytes	List of DEVICE_IDs. The number of DEVICE_IDs received would be less than or equal to MAXID specified in the GET_DEVICE_HANDLE Command . See Section 3.2 for a detailed description of DEVICE_ID.

8.9 SATA_COMPLETION Response

Description

This message is sent by the SPC 8x6G controller to notify a completion of a SATA request previously issued to the SPC 8x6G by the [SATA_HOST_IO_START Command](#) (Section 7.16) and the [SATA_ABORT Command](#) (Section 7.17).

Usage

Initiator.

Response Format

Table 138 SATA_COMPLETION Response Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=n	Reserved	OBID	CAT=0x02	OPC=0x00d
1	HTAG							
2	STATUS							
3	PARAM							
4	SATA Response FIS – If exists (SATARSP)							
...								
N								

Table 139 SATA_COMPLETION Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
0 [3:0]	Buffer Count	BC	5 bits	Buffer count. The SATA_COMPLETION Response can use more than one buffer/message entry depending on whether if MORE RESP FIS is reported.
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SATA_HOST_IO_START Command .

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	SATA Completion Status	STATUS	4 Bytes	<p>Status of the SATA operation.</p> <p>0x00000000: IO_COMPLETED. An I/O request completed successfully. If the PARAM field is zero, the SATA request is successfully completed and SATARSP is not valid. No other action is needed. If the PARAM field is not zero, it represents the length (in bytes) of the valid bytes in SATARSP.</p> <p>0x00000001: IO_ABORTED. An I/O aborted successfully. The PARAM contains additional information:</p> <ul style="list-style-type: none"> 0: Aborted by an abort command. Other values: Reserved. Other parameters are not used. <p>0x00000003: IO_UNDERFLOW. An I/O had a data underrun. The PARAM field indicates the residual bytes that were not transferred.</p> <p>0x00000007: IO_NO_DEVICE. An I/O request was sent to a non-existent device (invalid DEVICE_ID). Other parameters are not used.</p> <p>0x0000000F: IO_XFER_ERR_PHY_NOT_READY. An I/O aborted due to a PHY_NOT_READY during connection when sending H2D FIS. The PARAM field is not used.</p> <p>0x00000010: IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED. An I/O request failed due to an open connection error when sending H2D FIS. The host must do a rediscovery. The PARAM field is not used.</p> <p>0x00000011: IO_OPEN_CNX_ERROR_ZONE_VIOLATION. An I/O request failed due to an open connection error when sending H2D FIS. The host must do a rediscovery. The PARAM field is not used.</p> <p>0x00000012: IO_OPEN_CNX_ERR_BREAK. An I/O request failed due to an open connection error because a BREAK was received when sending H2D FIS. The PARAM field is not used.</p> <p>0x00000013: IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. An I/O request failed due to an open connection error when sending H2D FIS. The SPC 8x6G will automatically set the device state to DS_NON_OPERATIONAL. The host is required to complete the recovery procedure as described in Section 11.5.6, “Device States Related Error Handler” and Section 11.5.6.3, “Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices”. The PARAM field is not used.</p> <p>0x00000014: IO_OPEN_CNX_ERROR_BAD_DESTINATION. An</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>I/O request failed due to an open connection error when sending H2D FIS. The PARAM field is not used.</p> <p>0x00000015: IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED. An I/O request failed due to an open connection error when sending H2D FIS. The PARAM field is not used.</p> <p>0x00000016: IO_OPEN_CNX_ERROR_STP_RESOURCES_BUSY. An I/O request failed due to an open connection with the STP busy error when sending H2D FIS. Other parameters are not used.</p> <p>0x00000017: IO_OPEN_CNX_ERROR_WRONG_DESTINATION. An I/O request failed due to an open connection error when sending H2D FIS. The PARAM field is not used.</p> <p>0x00000019: IO_XFER_ERROR_NAK_RECEIVED. The SPC 8x6G receives an R_ERR from the SATA device when sending H2D FIS.</p> <p>0x0000001D: IO_XFER_ERROR_DMA. An I/O aborted due to an error previously detected in the data transfer phase (e.g. data overflow when SATA_EVENT of IO_OVERFLOW was previously posted for this IO). The PARAM field is not used.</p> <p>0x0000001F: IO_XFER_ERROR_SATA_LINK_TIMEOUT. An I/O request failed due to one of the following conditions: <ul style="list-style-type: none"> • A SYNC is received during H2D FIS transmission • H2D FIS transmission or reception is taking longer than the specified STP_FRAME_TIMEOUT value described in Section 10.5.3, “Timer Control 2 Register”. Other parameters are not used. </p> <p>0x00000021: IO_XFER_ERROR_REJECTED_NCQ_MODE. Returned if a new I/O is issued to a SATA device that is currently in an NCQ error condition state. The NCQ error condition has been previously reported through a SATA_EVENT Notification (Section 8.10) with EVENT set to IO_XFER_ERROR_ABORTED_NCQ_MODE.</p> <p>0x00000024: IO_XFER_OPEN_RETRY_TIMEOUT. An I/O aborted due to a open retry TIMEOUT when sending H2D FIS. The PARAM field is not used.</p> <p>0x00000038: IO_PORT_IN_RESET. An I/O aborted due to the port in the reset state. See Section 11.5.2, “Local PHY Control Link/Hard Reset Handler (Host Initiated)” for the port reset state. The PARAM</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>field is not used.</p> <p>0x00000039: IO_DS_NON_OPERATIONAL. An I/O aborted due to the device state in the DS_NON_OPERATIONAL state. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states. The PARAM field is not used.</p> <p>0x0000003A: IO_DS_IN_RECOVERY. An I/O aborted due to the device state in the DS_IN_RECOVERY state. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states. The PARAM field is not used.</p> <p>0x0000003E: IO_DS_IN_ERROR. An I/O aborted due to the device state in the DS_IN_ERROR state. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states.</p> <p>0x0000003F: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY. An I/O request failed due to an open connection error. The SPC 8x6G cannot allocate the hardware resource when it tries to open a connection to the target. Host needs to retry that I/O request.</p> <p>0x00000043: IO_OPEN_CNX_ERROR_HW_RESOURCE_BUSY_ALT. An I/O request failed due to an open connection error. The SPC 8x6G drains the I/O to prevent a lockup. See Section 3.24.1.</p>
3	SATA Completion Parameters	PARAM	4 Bytes	Additional information. The definition of this field depends on the STATUS field.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4..N	SATA Response FIS	SATARSP	(N-3) * 4 Bytes	<p>SATA response FIS. Only valid if STATUS is IO_COMPLETED (0x00000000) and PARAM field is not zero.</p> <p>Little Endian format. See Section 2.1.3.2, “SATA Payload Endianness” for details about SATA payload endianness.</p> <p>The type of FIS depends on the type of SATA protocol used (PIO, DMA, FPDMA or non-data):</p> <ul style="list-style-type: none"> PIO Read: In the case of RETFIS=1 and no error detected, nothing is returned. In the case of an error: TBD. PIO Write: Device To Host FIS received from the device. DMA Read: Device To Host FIS received from the device. DMA Write: Device To Host FIS received from the device. FPDMA Read: Set Device Bit FIS received from the device. FPDMA Write: Set Device Bit FIS received from the device. Non-Data: Device To Host FIS received from the device.

8.10 SATA_EVENT Notification

Description

This event is sent by the SPC 8x6G controller to notify the host of an event associated with a SATA port or a SATA device. The SPC 8x6G sends a SATA_EVENT notification to indicate that the I/O is not finished (may be pending in the SPC 8x6G or the remote target) and that the host must take additional action. The host action required is dependent on the EVENT received.

The event includes any errors from the SATA drive during an NCQ I/O operation (EVENT is IO_XFER_ERROR_ABORTED_NCQ_MODE). In response to this event, the SATA host driver needs to send a READ LOG EXT page 0x10 ATA command to interrogate the detailed SATA device error. Any pending I/Os to the same SATA drive must be aborted using the [SATA_ABORT Command](#) described in Section 7.17.

There are other events that the SPC 8x6G detects. For these, the SPC 8x6G notifies the host of errors and the pending commands on the SATA drive. The host must reset the SATA drive and abort the pending I/O in the SPC 8x6G by sending a [SATA_ABORT Command](#).

Usage

Initiator.

Event Format

Table 140 SATA Event Format

	Byte 3			Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1			Reserved	OBID	CAT=0x02	OPC=0x00e		
1	HTAG											
2	EVENT											
3	Reserved								PORT ID			
4	DEVICE_ID											
5	Reserved											
...												
15												

Table 141 SATA Event Fields

DWord	Field	Length	Description
1	HTAG	4 Bytes	When EVENT is set to IO_XFER_ERROR_ABORTED_NCQ_MODE , HTAG is not valid. For other EVENT, this is the HTAG passed in the SATA_HOST_IO_START Command described in Section 7.16.
2	EVENT	4 Bytes	Type of event: 0x00000002: IO_OVERFLOW. The SPC 8x6G detects an overflow during the data transfer phase. The I/O is not aborted. The HTAG field represents the TAG originally passed in the SATA_HOST_IO_START Command described in Section 7.16. 0x0000000E: IO_XFER_ERROR_BREAK. The SPC 8x6G detects a BREAK during connection after the command has been sent. This command may be pending in the SATA drive. 0x0000000F: IO_XFER_ERR_PHY_NOT_READY. The SPC 8x6G detects a PHY disruption during the connection after the command has been sent. This command may be pending in the SATA drive. 0x00000010: IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED. The SPC 8x6G could not open a connection with the STP target during the data phase. This command is still pending in the SATA drive. 0x00000011: IO_OPEN_CNX_ERROR_ZONE_VIOLATION. The SPC 8x6G could not open a connection with the STP

DWord	Field	Length	Description
			<p>target during the data phase. This command is still pending in the SATA drive.</p> <p>0x00000012: IO_OPEN_CNX_ERROR_BREAK. The SPC 8x6G could not open a connection with the STP target during the data phase. This command is still pending in the SATA drive.</p> <p>0x00000013: IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. The SPC 8x6G could not open a connection with the STP target during the data phase. The SPC 8x6G will automatically set the device state to DS_NON_OPERATIONAL. The host is required to complete the recovery procedure as described in Section 11.5.6, “Device States Related Error Handler” and Section 11.5.6.3, “Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices”. This command is still pending in the SATA drive.</p> <p>0x00000014: IO_OPEN_CNX_ERROR_BAD_DESTINATION. Only as initiator, the SPC 8x6G could not open a connection with the STP target during the data phase. This command is still pending in the SATA drive.</p> <p>0x00000015: IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED. The SPC 8x6G could not open a connection with the STP target during the data phase. This command is still pending in the SATA drive.</p> <p>0x00000016: IO_OPEN_CNX_ERROR_STP_RESOURCES_BUSY. The SPC 8x6G could not open a connection with the STP target during the data phase due to an STP affiliation problem in the STP bridge device (STP_RESOURCE_BUSY). This command is still pending in the SATA drive.</p> <p>0x00000017: IO_OPEN_CNX_ERROR_WRONG_DESTINATION. Only as initiator, the SPC 8x6G could not open a connection with the STP target during the data phase. This command is still pending in the SATA drive.</p> <p>0x00000019: IO_XFER_ERROR_NAK RECEIVED. The SPC 8x6G receives an R_ERR from the target during the data phase. This command is still pending in the target.</p> <p>0x0000001B: IO_XFER_ERROR_PEER_ABORTED. The SPC 8x6G has received an error due to one of the following conditions: <ul style="list-style-type: none"> • A SYNC is received during data FIS transmission. • Frame data transmission or reception is taking longer than the specified STP_FRAME_TIMEOUT value described in Section 10.5.3, “Timer Control 2 Register”. This command may be pending in the SATA drive.</p> <p>0x00000020: OSSA_IO_XFER_ERROR_SATA. An SATA/STP io aborted due to a zero transfer length or a SATA/STP link layer parity error is detected by the controller.</p>

DWord	Field	Length	Description
			<p>The PARAM field is not used.</p> <p>0x00000023: IO_XFER_ERROR_ABORTED_NCQ_MODE. The SPC 8x6G controller has received an error from the SATA drive during an NCQ I/O.</p> <p>In response to this event, the host must send a READ LOG EXT page 0x10 ATA command to obtain detailed error information. The SPC 8x6G rejects any command other than READ LOG EXT command when this condition occurs. The HTAG field is not valid. Any pending I/Os in the SPC 8x6G controller to the affected SATA drive must be aborted using the SATA_ABORT Command.</p> <p>0x00000024: IO_XFER_OPEN_RETRY_TIMEOUT. The SPC 8x6G failed to open a connection with the target after a number of retries. This command is still pending in the target.</p> <p>0x00000026: IO_XFER_ERROR_UNEXPECTED_PHASE. The SPC 8x6G receives an unexpected frame when:</p> <ul style="list-style-type: none"> • The SPC 8x6G is receiving data when it is programmed to send data • The SPC 8x6G is receiving DMA/PIO Setup FIS with D Bit=0 when it is programmed to receive data. <p>This command is still pending in the target.</p> <p>0x00000027: IO_XFER_ERROR_XFER_RDY_OVERRUN. The SPC 8x6G receives a length in the DMA SETUP FIS field larger than the transfer length it is programmed to send. This command is still pending in the target.</p> <p>0x00000028: IO_XFER_ERROR_XFER_RDY_NOT_EXPECTED. The SPC 8x6G receives data when it is not expecting any, for example, the SPC 8x6G is receiving DMA setup with the D bit set to zero for a read command. This command is still pending in the target.</p> <p>0x00000034: IO_XFER_ERROR_OFFSET_MISMATCH. The SPC 8x6G detects a DATA phase offset mismatch during the data phase. This command is still pending in the SATA drive.</p> <p>0x00000035: IO_XFER_ERROR_XFER_ZERO_DATA_LEN. The SPC 8x6G detects SATA mode and DMA or PIO setup FIS is received with a DMA transfer count of zero. This command is still pending in the target. The host may need to reset the SATA drive and abort the pending I/O in the SPC 8x6G by sending a SMP_ABORT Command.</p> <p>0x00000036: IO_XFER_CMD_FRAME_ISSUED. The SPC 8x6G has sent the frame on the wire and an R_OK primitive has been received. This is not an error. This event is only reported if the host set the 'M' field in the SATA_HOST_IO_START Command.</p> <p>0x0000003C: IO_XFER_PIO_SETUP_ERROR. An I/O aborted due to an error in the PIO Setup FIS.</p>
3 [3:0]	PORT ID	4 bits	Identification for an SPC 8x6G local SATA port context.

DWord	Field	Length	Description
4	DEVICE_ID	4 Bytes	Device identification. See Section 3.2 for a detailed description of DEVICE_ID.

8.11 SSP_EVENT Notification

Description

This event is sent by the SPC 8x6G controller to notify the host about an error detected while processing an SSP request previously issued to the controller by the [SSP_INI_IO_START Command](#) described in Section 7.4.

All event codes apply to both Initiator and Target modes unless otherwise specified.

The SPC 8x6G sends this SSP_EVENT notification to indicate that the I/O is not finished (may be pending in SPC 8x6G or the remote SSP device) and that the host must take additional action. The host action required is depending on the EVENT received.

The event also indicates that the COMMAND may or may not have arrived and be pending at the target. This includes an ACK/NAK timeout for a COMMAND frame (which will be reported with IO_XFER_ERROR_CMD_ISSUE_ACK_NAK_TIMEOUT). The host may send a QUERY_TASK task management request to determine whether the COMMAND needs to be resent or not. The host may also decide to send an ABORT_TASK task management request to the target device and, at its completion, an [SSP_ABORT Command](#) (Section 7.10) to clean up the corresponding pending I/O in the SPC 8x6G.

There are also events indicating that a COMMAND is still pending at the target. These events include the detection of an error from opening a connection with the target during the data phase (will be reported with IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS). The host may also decide to send an ABORT_TASK task management command to the target device and, at its completion, an [SSP_ABORT Command](#) to clean up the corresponding pending I/O in the SPC 8x6G.

Usage

Initiator.

Event Format

Table 142 SSP Event Format

	Byte 3				Byte 2		Byte 1		Byte 0							
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x00f								
1	HTAG															
2	EVENT															
3	Reserved															
4	DEVICE_ID															
5	Reserved				SSP TAG											
6	Reserved															
...																
15																

Table 143 SSP Event Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SSP_XX_XX_START Command.
2	Event Type	EVENT	4 Bytes	Type of event: 0x00000002: IO_OVERFLOW. Only as initiator, the SPC 8x6G detects an overflow during data transfer phase. I/O is not aborted. 0x0000000E: IO_XFER_ERROR_BREAK. Only as initiator, the SPC 8x6G detects BREAK during connection after the command has been sent. This command may be pending in the target. 0x0000000F: IO_XFER_ERR_PHY_NOT_READY. Only as initiator, the SPC 8x6G detects PHY disruption during connection after the command has been sent. This command may be pending in the target. 0x00000010: IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED. Only as initiator, the SPC 8x6G could not open a connection with the target during the data phase. This command is still pending in the target. 0x00000011: IO_OPEN_CNX_ERROR_ZONE_VIOLATION. Only as initiator, the SPC 8x6G could not open a connection with the target during the data phase. This command is still pending in the target. 0x00000012: IO_OPEN_CNX_ERROR_BREAK. Only as initiator, the SPC 8x6G could not open a connection with the target during the data phase.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>This command is still pending in the target.</p> <p>0x00000013: IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. The SPC 8x6G could not open a connection during the D phase. The SPC 8x6G will automatically set the device state to DS_NON_OPERATIONAL. The host is required to complete the recovery procedure as described in Section 11.5.6, "Device States Related Error Handler" and Section 11.5.6.3, "Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices". The PARAM field is not used.</p> <p>0x00000014: IO_OPEN_CNX_ERROR_BAD_DESTINATION. Only as initiator, the SPC 8x6G could not open a connection with the target during the data phase. This command is still pending in the target.</p> <p>0x00000015: IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED. Only as initiator, the SPC 8x6G could not open a connection with the target during the data phase. This command is still pending in the target.</p> <p>0x00000017: IO_OPEN_CNX_ERROR_WRONG_DESTINATION. Only as initiator, the SPC 8x6G could not open a connection with the target during the data phase. This command is still pending in the target.</p> <p>0x00000019: IO_XFER_ERROR_NAK RECEIVED. Only as initiator, the SPC 8x6G detects NAK from the target during the data phase. This command is still pending in the target.</p> <p>0x0000001A: IO_XFER_ERROR_ACK_NAK_TIMEOUT. As initiator, the SPC 8x6G detects an ACK/NAK timeout during the data phase. This command is still pending in the target.</p> <p>0x00000024: IO_XFER_OPEN_RETRY_TIMEOUT: As initiator, the SPC 8x6G failed to open a connection with the target after a number of retries. This command is still pending in the target. As target, the SPC 8x6G failed to open a connection with the initiator after a number of retries. The HTAG field represents the TAG originally passed in the SSP_TGT_IO_START Command.</p> <p>0x00000026: IO_XFER_ERROR_UNEXPECTED_PHASE. In initiator only, the SPC 8x6G receives an unexpected frame when: <ul style="list-style-type: none"> • The SPC 8x6G is receiving data when it </p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>is programmed to send data.</p> <ul style="list-style-type: none"> The SPC 8x6G is receiving XFER_RDY frames when it is programmed to receive data. <p>This command is still pending in the target.</p> <p>0x00000027: IO_XFER_ERROR_XFER_RDY_OVERRUN. As initiator, the SPC 8x6G receives a XFER_RDY request with a WRITE DATA LENGTH field larger than the transfer length it is programmed to send. This command is still pending in the target.</p> <p>0x00000028: IO_XFER_ERROR_XFER_RDY_NOT_EXPECTED. As initiator, the SPC 8x6G receives a XFER_RDY when it is programmed to receive data. This command is still pending in the target.</p> <p>0x00000030: IO_XFER_ERROR_CMD_ISSUE_ACK_NAK_TIMEOUT. Only as initiator, the SPC 8x6G detects an ACK/NAK timeout during the command phase. This command may or may not be pending in the target.</p> <p>0x00000034: IO_XFER_ERROR_OFFSET_MISMATCH. In Initiator mode, the SPC 8x6G detects, during a SCSI READ operation, that the DATA_OFFSET field in the SSP frame header is out of sequence, or during a SCSI WRITE operation, that the REQUESTED_OFFSET field in the XFER_RDY frame is out of sequence. This command is still pending in the target.</p> <p>0x00000035: IO_XFER_ERROR_XFER_ZERO_DATA_LEN. Only as initiator, the SPC 8x6G detects XFER_RDY request with zero transfer size. This command is still pending in the target.</p> <p>0x00000036: IO_XFER_CMD_FRAME_ISSUED. Only as initiator, the SPC 8x6G has sent the frame on the wire and ACK primitive has been received. This is not an error. This event is only reported if the host set the 'M' field in the SSP_INI_IO_START Command, the SSP_INI_TM_START Command or the SSP_INI_EXT_IO_START Command. The HTAG field represents the TAG originally passed in above commands.</p>
3 [3:0]	Port Identification	PORt ID	4 bits	Identification for an SPC 8x6G local SAS port context.
4	Device Identification	DEVICE_ID	4 Bytes	Device identification. See Section 3.2 for a detailed description of DEVICE_ID.
5	SSP Initiator	SSP TAG	2 Bytes	The SSP (Initiator) TAG defined as the TAG field

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
	Tag			in bytes 16 and 17 of the SSP frame. This field is used for debugging purposes, that is, matching SSP completion with the SAS trace in the SAS analyzer.

8.12 DEVICE_HANDLE_ARRIVED Notification

Description

This event is sent by the SPC 8x6G controller in target mode to notify the host of the arrival of a new initiator device handle. The device handle represents the SCSI nexus of the local target port to the remote initiator port.

From an MPI perspective, the firmware and the host exchange the device handle information using a DEVICE_ID field defined as a 16-bit index. A DEVICE_ID field indirectly represents a device handle.

The host can either accept or reject the device handle.

Accepting the device handle implies that the host responds to the remote initiator by responding to the frames received from this initiator. Rejecting the device handle implies that the host does not respond to the frames received from this initiator (that is, it silently drops frames from this initiator).

The host accepts or rejects the initiator by sending the [DEVICE_HANDLE_ACCEPT Command](#) described in Section 7.7.

Usage

Target.

Event Format

Table 144 DEVICE_HANDLE_ARRIVED Event Format

	Byte 3				Byte 2		Byte 1		Byte 0						
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0010							
1	CTAG														
2	DEVICE_ID														
3	Reserved				PROT		CONR	PORT_ID							
4	SASAH														
5	SASAL														
6	Reserved														
...															
15															

Table 145 DEVICE_HANDLE_ARRIVED Event Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Controller Firmware TAG	CTAG	4 Bytes	Tag or context for this operation. The tag value is assigned by firmware.
2	Device Identifier	DEVICE_ID	4 Bytes	Device Identification of the initiator. See Section 3.2 for a detailed description of DEVICE_ID.
3 [10:8]	Protocol	PROT	3 bits	The PROTOCOL field of the received Open Address frame: 000b: SMP 001b: SSP All others reserved.
3 [7:4]	Connection Rate	CONR	4 bits	The CONNECTION RATE field of the received Open Address frame: 0x08: 1.5 Gbit/s 0x09: 3.0 Gbit/s 0x0A: 6.0 Gbit/s All others reserved.
3 [3:0]	Port Identifier	PORT_ID	4 bits	Identification for the SPC 8x6G local SAS/SATA port.
4	SAS Address High	SASAH	4 Bytes	High 32 bits of the remote initiator SAS address in Big Endian format.
5	SAS Address Low	SASAL	4 Bytes	Low 32 bits of the remote initiator SAS address in Big Endian format.

8.13 SSP_REQUEST RECEIVED Notification

Description

This event is sent by the SPC 8x6G controller to the host to indicate the arrival of an SSP request (SSP command frame, task management frame, or a vendor-specific frame).

In target mode, the host initiates the data transfer using the [SSP_TGT_IO_START Command](#) described in Section [7.8](#).

In target mode, the host sends the SSP response using the [SSP_TGT_RESPONSE_START Command](#) described in Section [7.9](#).

Usage

Target.

Event Format

Table 146 SSP_REQUEST RECEIVED Event Format

	Byte 3			Byte 2		Byte 1		Byte 0									
0	V	H	R	BC=n	Reserved	OBID	CAT=0x02	OPC=0x0012									
1	DEVICE_ID																
2	INI_TAG				SSPIUL												
3	FRAME_TYPE		HSSA														
4	Reserved		TLR	HDSA													
5	SSP IU ((M-3) * 4 bytes)																
...																	
M																	

Table 147 SSP_REQUEST RECEIVED Event Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Device Identifier	DEVICE_ID	4 Bytes	The device identification of the initiator.
2 [31:16]	SSP Frame Initiator Tag	INI_TAG	2 Bytes	The SSP TAG value that the initiator port set. This is bytes 16 and 17 of the SSP frame.
2 [15:0]	SSP Information Unit Length	SSPIUL	2 Bytes	The length in bytes of the SSP IU.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
3 [31:24]	SSP Frame Type	FRAME TYPE	1 Bytes	The type as defined in byte 0 of the SSP frame: 0x06: COMMAND frame 0x16: TASK frame 0xF0-0xFF: Vendor specific
3 [23:0]	HASHED SOURCE SAS ADDRESS	HSSA	3 Bytes	HASHED SOURCE SAS ADDRESS from the SSP frame received. This field is provided for debug purposes only.
4 [25:24]	Transport Layer Retry Control	TLR	2 bits	This field contains the TLR Control field from the SSP frame from the initiator as defined in the SAS-2 specification.
4 [23:0]	HASHED DESTINATION SAS ADDRESS	HDSA	3 Bytes	HASHED DESTINATION SAS ADDRESS from the SSP frame received. This field is provided for debug purposes only.
5... M	SSP Information Unit	SSPIU	(M-4) * 4 Bytes	SSP IU: Command IU or task management Function IU depending on FRAME TYPE. Big Endian format. See Section 2.1.3.1, “ SAS Payload Endianness ” for details about SAS payload endianness.

8.14 GET_DEVICE_INFO Response

Description

This message is sent by the SPC 8x6G controller as the response to the [GET_DEVICE_INFO Command](#) described in Section 7.19.

Usage

Initiator and target.

Response Format

Table 148 GET_DEVICE_INFO Response Format

	Byte 3				Byte 2		Byte 1		Byte 0										
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x0013											
1	HTAG																		
2	STATUS																		
3	DEVICE_ID																		
4	Reser ved	S	DLR	SMPT					Reser ved	A	R	PORT ID							
5	Reserved					ITNT													
6	SADDRH																		
7	SADDRL																		
8	Reserved																		
...	Reserved																		
15																			

Table 149 GET_DEVICE_INFO Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the GET_DEVICE_INFO Command .
2	Get Device Information Completion Status	STATUS	4 Bytes	Status of the operation: 0x00000000: SUCCESS. 0x00000003: FAILURE. The rest of device information fields (DWord [15:3] are not valid.
3	Device Identifier	DEVICE_ID	4 Bytes	Device Identifier. See Section 3.2 for a detailed description of DEVICE_ID.
4 [29:28]	SAS or SATA Device Type	S	2 bits	Two bit S flag to specify device type: 00b: STP device 01b: SSP or SMP device 10b: Directly-attached SATA device
4 [27:24]	Device Link Rate	DLR	4 bits	The controller uses this value (of the Open Address frame) for the Connection Rate field when opening the device: 0x08: 1.5 Gbit/s. 0x09: 3.0 Gbit/s. 0x0A: 6.0 Gbit/s. All others reserved.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
4 [23:8]	SMP Timeout	SMPT	2 Bytes	The time in 20 µs units where the controller waits for an SMP response from an SMP target. If the timer expires before the arrival of the SMP response, the SPC 8x6G sends a BREAK and an error message is posted.
4 [5]	AWT flag	A	1 bit	Priority setting for AWT (Arbitration Wait Time) for this device: 0b: Default setting (recommended). The actual AWT value is based on how long an OPEN frame has been waiting for a connection request to be accepted. It starts at 0. As specified in the SAS specification, from 0 to 32768 µs, the AWT value is incremented every µs. From 32768 µs and on, the AWT value is incremented every ms. 1b: Increase priority. The actual AWT value starts at 32768 µs when the A flag is set to '1'. The AWT value is incremented every ms from that point on.
4 [4]	Retry flag	R	1 bit	This field indicates whether the Transport Layer Retry (TLR) flag per SAS 1.1 and SAS 2.0 is enabled: 1b: Enable TLR 0b: Disable TLR See Section 3.20, "Transport Layer Retry (TLR) Handling".
4 [3:0]	Port Identifier	PORT_ID	4 bits	The ID for the SPC 8x6G controller's local SAS/SATA port context from where this device is accessible.
5 [31:16]	Reserved	Reserved	2 Bytes	Reserved
5 [15:0]	IT Nexus Timeout	ITNT	2 Bytes	The value in milliseconds of the time unit that is used by the controller to determine the nexus timeout condition.
6	SAS Address High	SADDRH	4 Bytes	High 32 bits of device SAS address. Not used for a directly attached SATA drive. Big Endian format. See Section 2.1.3.1, "SAS Payload Endianness" for more details.
7	SAS Address Low	SADDRL	4 Bytes	Low 32 bits of device SAS address. Not used for a directly attached SATA drive. Big Endian format. See Section 2.1.3.1, "SAS Payload Endianness" for more details.

8.15 FW_FLASH_UPDATE Response

Description

This message is sent by the SPC 8x6G controller as the response to the [FW_FLASH_UPDATE Command](#) described in Section 7.20.

Usage

Initiator and target.

Response Format

Table 150 FW_FLASH_UPDATE Response Format

	Byte 3			Byte 2		Byte 1		Byte 0			
0	V	H	R	BC=1			Reserved	OBID	CAT=0x02	OPC=0x014	
1	HTAG										
2	STATUS										
3	Reserved										
...											
15											

Table 151 FW_FLASH_UPDATE Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the FW_FLASH_UPDATE Command .
2	Flash update status	STATUS	4 Bytes	<p>0x00000000: FLASH_UPDATE_COMPLETE_PENDING_REBOOT.</p> <p>Firmware flash update is completed successfully and is waiting for reboot to take into effect.</p> <p>0x00000001: FLASH_UPDATE_IN_PROGRESS.</p> <p>When firmware flash update is done in piecewise blocks, this status indicates that the previous FW_FLASH_UPDATE Command was successful and firmware is ready to receive the next FW_FLASH_UPDATE Command.</p> <p>0x00000002: FLASH_UPDATE_HDR_ERR.</p> <p>Error in image header.</p> <p>0x00000003: FLASH_UPDATE_OFFSET_ERR.</p> <p>Offset value in current IOMB is out of sequence.</p> <p>0x00000004: FLASH_UPDATE_CRC_ERR.</p> <p>Computed CRC did not match with header CRC.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>0x00000005: FLASH_UPDATE_LENGTH_ERR. Image length mismatch with image header.</p> <p>0x00000006: FLASH_UPDATE_HW_ERR. Error programming the flash memory – firmware flash update aborted.</p> <p>0x00000010: FLASH_UPDATE_DNLD_NOT_SUPPORTED. In current firmware state (for example, HDA mode), a firmware download is not supported.</p> <p>0x00000011: FLASH_UPDATE_DISABLED. Flash download is disabled in the initialization string.</p> <p>0x00000012: FWDNLD_DEVICE_UNSUPORT. The flash device does not exist or is not supported.</p>

8.16 GPIO Response

Description

This response is sent by the SPC 8x6G controller as the response to the [GPIO Command](#) described in Section [7.21](#).

Usage

Initiator and target.

Response Format

Table 152 GPIO Response Format

	Byte 3				Byte 2			Byte 1				Byte 0														
0	V	H	R	BC=1	Reserved	OBID		CAT=0x02			OPC=0x016															
1	HTAG																									
2	Reserved																									
3	Reserved																									
4	Reserved				GPIORDVAL																					
5	Reserved				GPIOIE																					
6	Reserved		OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0												
7	Reserved				Reserved	OT19	OT18	OT17	OT16	OT15	OT14	OT13	OT12													
8	Reserved				GPIOEVCHANGE																					
9	Reserved				GPIOEVRISE																					
10	Reserved				GPIOEVFALL																					
11	Reserved																									
...																										
15																										

Table 153 GPIO Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	The context specified in the GPIO Command .
4 [19:0]	GPIO Read Value	GPIORDVAL	20 bits	When the GR field in the GPIO Command is 1b, this field is a bit map containing the corresponding value for each GPIO pin. If the corresponding GPIO pin input signal is enabled, the value reflects the logical AND of level driven from the external device and the previously written GPIO signal. When the GPIO pin input is not enabled, the value is not valid.
5 [19:0]	GPIO Input Enabled	GPIOIE	20 bits	When the GR field in the GPIO Command is 1b, this field is a bitmap that contains the information whether the input is enabled for the corresponding GPIO pin. 0b: The corresponding GPIO pin is not input enabled. Output is always enabled. 1b: The corresponding GPIO pin is input enabled. Output is always enabled.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
6 [23:0]	GPIO Type	OT[0..11]	12x2 bits	When the GR field in the GPIO Command is 1b (for GPIO signals 0 to 11), if the corresponding bit in the GPIOIE field is set to 0b, this field contains the output signal type: 00b: Tristated (default) 01b: Driven 10b: Reserved 11b: Reserved
7 [15:0]	GPIO Type	OT[12..19]	8x2 bits	When the GR field in the GPIO Command is 1b, for GPIO signals 12 to 19, if the corresponding bit in the GPIOIE field is set to 0b, this field contains the output signal type. 00b: Tristated 01b: Driven 10b: Reserved 11b: Reserved
8 [19:0]	GPIO Event Level	GPIOEVCHAN GE	20 bits	When the GR bit is set to 1b, this field is a bitmap that enables GPIO a event: 0b: A level change does not trigger a GPIO event 1b: A level change triggers a GPIO event
9 [19:0]	GPIO Event Rising Edge	GPIOEVRISE	20 bits	When the GR bit is set to 1b, this field is a bitmap that enables a GPIO event : 0b: Rising edge does not trigger a GPIO event 1b: Rising edge triggers a GPIO event
10 [19:0]	GPIO Event Falling Edge	GPIOEVFALL	20 bits	When the GR bit is set to 1b, this field is a bitmap that enables a GPIO event: 0b: Falling edge does not trigger a GPIO event 1b: Falling edge triggers a GPIO event

8.17 GPIO_EVENT Notification

Description

This GPIO event is sent when the GPIO events are enabled and the programmed GPIO condition occurs. (See Section 7.21, “[GPIO Command](#)”.)

Usage

Initiator and target.

Response Format

Table 154 GPIO Response Format

	Byte 3				Byte 2		Byte 1		Byte 0							
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x017								
1	Reserved				GPIOEVENT											
2	Reserved															
...																
15																

Table 155 GPIO Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1 [19:0]	GPIO Event	GPIOEVENT	20 bits	This field is a bit map that indicates which GPIO input pins have generated the event.

8.18 GENERAL_EVENT Notification

Description

This GENERAL_EVENT Notification is sent to inform the host of some general error related to a specific inbound IOMB.

This event includes the reporting of inbound IOMB with the Valid (V) bit not set in the IOMB header, unsupported IOMB operation invalid OBID set in the inbound IOMB.

The SPC 8x6G will send this event notification to the host using the outbound high priority path. See Section 3.18, “[High Priority Operation and Normal Priority IOMB Processing](#)”.

Usage

Initiator and target.

Event Format

Table 156 GENERAL_EVENT Notification Format

	Byte 3				Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x018	
1	STATUS								
2	INBOUND IOMB								
...									
15									

Table 157 GENERAL_EVENT Notification Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	STATUS	STATUS	4 bytes	0x00000001: INBOUND_IOMB_V_BIT_NOT_SET Inbound IOMB is received with the V bit in the IOMB header not set. 0x00000002: INBOUND_IOMB_OPC_NOT_SUPPORTED Inbound IOMB is received with an unsupported OPC. 0x00000003: INBOUND_IOMB_INVALID_OBID Inbound IOMB is received with an invalid OBID.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2:15	INBOUND IOMB	INBOUND IOMB	14 DWords (56 bytes)	The first 14 DWords (56 bytes) of the inbound IOMB which correspond to this event.

8.19 SSP_ABORT Response

Description

The SPC 8x6G controller sends this response as the response to the [SSP_ABORT Command](#) described in Section [7.10](#).

The I/O to be aborted itself, if found, will be reported in an [SSP_COMPLETION Response](#) with STATUS set to IO_ABORTED. See Section [8.3](#).

Usage

Initiator and target.

Response Format

Table 158 SSP_ABORT Response Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x01A
1	HTAG							
2	STATUS							
3	Reserved							SCP
4	Reserved							
...								
15								

Table 159 SSP_ABORT Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SSP_ABORT Command .
2	Response Status	STATUS	4 Bytes	<p>Status of the operation:</p> <p>0x00000000: IO_COMPLETED. The abort completed successfully.</p> <p>SCP Encoding Definition</p> <p>00b: The I/O to aborted (HTAG-ABT in SSP_ABORT Command) has been reported in an SSP_COMPLETION Response with STATUS set to IO_ABORTED.</p> <p>01bL All SSP I/Os associated with a device handle (DEVICE_ID in SSP_ABORT Command) have been reported in SSP_COMPLETION Response with STATUS set to IO_ABORTED.</p> <p>0x00000006: IO_NOT_VALID.</p> <p>SCP Encoding Definition</p> <p>00b: The SPC 8x6G detected that there is no valid I/O associated with the HTAG-ABT passed in the SSP_ABORT Command.</p> <p>01b: The SPC 8x6G detected that there is no valid device associated with the DEVICE_ID passed in the SSP_ABORT Command.</p> <p>0x00000040: IO_ABORT_IN_PROGRESS.</p> <p>SCP Encoding Definition</p> <p>00b: The I/O specified by the HTAG-ABT is being aborted by another SSP_ABORT Command.</p> <p>01b: All the I/Os for the device are being aborted by another SSP_ABORT Command.</p>
3[1:0]	Scope of Abort Flag	SCP	2 bits	<p>The scope of abort flag previously set in SSP_ABORT Command:</p> <p>00b: Abort a single SSP I/O.</p> <p>01b: Abort all SSP I/Os for a DEVICE_ID.</p>

8.20 SATA_ABORT Response

Description

The SPC 8x6G controller sends this response as the response to the [SATA_ABORT Command](#) described in Section [7.17](#).

The I/O to be aborted itself, if found, will be reported in **SATA_COMPLETION** Response with STATUS set to IO_ABORTED. See Section 8.9.

Usage

Initiator and target.

Response Format

Table 160 SATA_ABORT Response Format

Table 161 SATA ABORT Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SATA_ABORT Command .

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2	Response Status	STATUS	4 Bytes	<p>Status of the operation:</p> <p>0x00000000: IO_COMPLETED. The abort completed successfully.</p> <p>SCP Encoding Definition</p> <p>00b: The I/O to aborted (HTAG-ABT in the SATA_ABORT Command) has been reported in SATA_COMPLETION Response with STATUS set to IO_ABORTED.</p> <p>01b: All I/Os associated with a device handle (DEVICE_ID in the SATA_ABORT Command) have been reported in the SATA_COMPLETION Response with STATUS set to IO_ABORTED.</p> <p>0x00000006: IO_NOT_VALID.</p> <p>SCP Encoding Definition</p> <p>00b: The SPC 8x6G detected that there is no valid I/O associated with the HTAG-ABT passed in the SATA_ABORT Command.</p> <p>01b: The SPC 8x6G detected that there is no valid device associated with the DEVICE_ID passed in the SATA_ABORT Command.</p> <p>0x00000040: IO_ABORT_IN_PROGRESS.</p> <p>SCP Encoding Definition</p> <p>00b: The I/O specified by HTAG-ABT is being aborted by another SATA_ABORT Command.</p> <p>01b: All the I/Os for the device are being aborted by another SATA_ABORT Command.</p>
3[1:0]	Scope of Abort Flag	SCP	2 bits	<p>The scope of abort flag previously set in SATA_ABORT Command:</p> <p>00b: Abort a single SATA I/O.</p> <p>01b: Abort all SATA I/Os for a DEVICE_ID.</p>

8.21 SAS_DIAG_MODE_START_END Response

Description

The SPC 8x6G controller sends this response as the response to the [SAS_DIAG_MODE_START_END Command](#) described in Section 7.22.

Usage

Initiator and target.

Response Format

Table 162 SAS_DIAG_MODE_START_END Response Format

	Byte 3				Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x01C			
1	HTAG										
2	STATUS										
3	Reserved										
...											
15											

Table 163 SAS_DIAG_MODE_START_END Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SAS_DIAG_MODE_START_END Command .
2	Status	STATUS		Status of the operation: 0x00: SUCCESS 0x01: PHY_INVALID 0x02: PHY_NOT_DISABLED 0x03: FAILURE

8.22 SAS_DIAG_EXECUTE Response

Description

The SPC 8x6G controller sends this response as the response to the [SAS_DIAG_EXECUTE Command](#) described in Section 7.23.

Usage

Initiator and target.

Response Format

Table 164 SAS_DIAG_EXECUTE Response Format

	Byte 3				Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x01D			
1	HTAG										
2	Reserved				CMD_TYPE	CMD_DESC	Reserved	PHYID			
3	STATUS										
4	REPORT_DATA										
5	Reserved										
...											
15											

Table 165 SAS_DIAG_EXECUTE Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SAS_DIAG_EXECUTE Command .
2 [15:13]	Command Type	CMD_TYPE	3 bits	Command type set in SAS_DIAG_EXECUTE Command .
2 [12:8]	Command Descriptor	CMD_DESC	5 bits	Command descriptor set in SAS_DIAG_EXECUTE Command .
2 [3:0]	PHY Identifier	PHYID	4 bits	Zero-based PHY Identifier.
3	Status	STATUS	4 Bytes	Status of the operation: 0x0000: SUCCESS 0x0001: INVALID_COMMAND 0x0002: FAIL 0x0003: NOT_IN_DIAGNOSTIC_MODE 0x0004: INVALID_PHY
4	Report Diagnostic Data	REPORT_DATA	4 Bytes	Valid only when CMD_TYPE is 100b: DIAG_REPORT_GET in SAS_DIAG_EXECUTE Command .

8.23 GET_TIME_STAMP Response

Description

The SPC 8x6G controller sends this response as the response to the [GET_TIME_STAMP Command](#) described in Section 7.25.

Usage

Initiator and target.

Response Format

Table 166 GET_TIME_STAMP Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02		
1	HTAG									
2	TIME STAMP LOWER									
3	TIME STAMP UPPER									
4										
...										
15										

Table 167 GET_TIME_STAMP Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the GET_TIME_STAMP Command .
2	Time stamp Lower	TIME STAMP LOWER	4 Bytes	SPC 8x6G lower 32-bit of internal time stamp associated with event log.
3	Time stamp Upper	TIME STAMP UPPER	4 Bytes	SPC 8x6G upper 32-bit of internal time stamp associated with event log.

8.24 SAS_HW_EVENT_ACK Response

Description

The SPC 8x6G controller sends this response as the response to the [SAS_HW_EVENT_ACK Command](#) described in Section 7.24.

Usage

Initiator and target.

Response Format

Table 168 SAS_HW_EVENT_ACK Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02		
1	HTAG									
2	STATUS									
3	Reserved									
...										
15										

Table 169 SAS_HW_EVENT_ACK Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SAS_HW_EVENT_ACK Command .
2	Status	STATUS	4 Bytes	Bit field status for the acknowledgement sent through SAS_HW_EVENT_ACK Command : All bits zero: successful event acknowledgement. Bit 0 set: Invalid SEA set in SAS_HW_EVENT_ACK Command . This check applies to IOP_EVENT_BROADCAST_CHANGE only. Bit 1 set: Invalid PHYID set in SAS_HW_EVENT_ACK Command . Bit 2 set : Invalid PORT_ID set in SAS_HW_EVENT_ACK Command . Bit 3 set: Invalid PARAM0 set in SAS_HW_EVENT_ACK Command . Other bits reserved.

8.25 PORT_CONTROL Response

Description

This response is sent by the SPC 8x6G controller as the response to the previous [PORT_CONTROL Command](#) described in Section [7.26](#).

Usage

Initiator and target.

Response Format

Table 170 PORT_CONTROL Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02 OPC=0x020		
1	HTAG									
2	Reserved			PORT_OP		Reserved	PORT_ID			
3	STATUS									
4	Reserved									
...										
15										

Table 171 PORT_CONTROL Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the PORT_CONTROL Command .
2 [3:0]	Port Identifier	PORT_ID	4 bits	PORT_ID previously passed in the PORT_CONTROL Command .
2 [15:8]	PORT Operation	PORT_OP	1 Byte	PORT Operation previously passed in the PORT_CONTROL Command .
3	Status	STATUS	4 Bytes	Status of the operation: 0x00: SUCCESS. The operation completed successfully. 0x03: FAILURE. The operation failed.

8.26 SKIP_ENTRIES_EVENT Notification

Description

The skip entries event is sent by the SPC 8x6G to allow the controller to advance the OQ PI. This is a no-option IOMB. Its main purpose is to advance the OQ PI.

A skip entry is typically done to allow the SPC 8x6G to pass multiple IOMB entries ($BC > 1$) to the host in contiguous memory without splitting the IOMB into non-contiguous memory when a wrap-around is about to occur.

The BC field in the IOMB header indicates the amount the host must increment the CI count.

Usage

Initiator and target.

Response Format

Table 172 SKIP_ENTRIES_EVENT Notification

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=n	Reserved	OBID	CAT=0x02	OPC=0x021
1								
...								
15								
					Reserved			

Table 173 SKIP_ENTRIES_EVENT Notification Field

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
-	-	-	-	-

8.27 SMP_ABORT Response

Description

The SPC 8x6G controller sends this response as the response to an [SMP_ABORT Command](#), which is described in Section [7.14](#).

If found, the I/O to be aborted will be reported in the [SMP_COMPLETION Response](#) with the status set to IO_ABORTED. See Section [8.4](#).

Usage

Initiator and Target

Response Target

Table 174 SMP_ABORT Response Format

	Byte 3			Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1			Reserved	OBID	CAT=0x02	OPC=0x022		
1	HTAG											
2	STATUS											
3	Reserved							SCP				
4	Reserved											
...												
15												

Table 175 SMP_ABORT Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SMP_ABORT Command .
2	Response Status	STATUS	4 Bytes	<p>Status of the operation: 0x00000000: IO_COMPLETED. The abort completed successfully. SCP Encoding Definition 00b: The I/O to aborted (HTAG-ABT in SMP_COMPLETION Response) has been reported in an SMP_COMPLETION Response with STATUS set to IO_ABORTED. 01bL All SSP I/Os associated with a device handle (DEVICE_ID in SMP_ABORT Command) have been reported in SMP_COMPLETION Response with STATUS set to IO_ABORTED.</p> <p>0x00000006: IO_NOT_VALID. The SPC 8x6G detected that there is no valid device associated with the DEVICE_ID passed in the SMP_ABORT Command). SCP Encoding Definition 00b: The SPC 8x6G detected that there is no valid I/O associated with the HTAG-ABT passed in the SMP_COMPLETION Response. 01b: The SPC 8x6G detected that there is no valid device associated with the DEVICE_ID passed in the SMP_COMPLETION Response.</p>
3 [1:0]	Scope of Abort Flag	SCP	2 bits	The scope of abort flag previously set in SMP_ABORT Command : 00b: Abort a single SMP I/O. 01b: Abort all SMP I/Os for a DEVICE_ID.

8.28 GET_NVMD_DATA Response

Description

This response is sent by the SPC 8x6G controller as the response to the [GET_NVMD_DATA Command](#) described in Section 7.27.

Usage

Initiator and target.

Response Format

Table 176 GET_NVMD_DATA Response Format

	Byte 3			Byte 2		Byte 1		Byte 0							
0	V	H	R	BC=1		Reserved	OBID	CAT=0x2	OPC=0x023						
1	HTAG														
2	IP	Reserved		TDA		TBN	TDPS	TDAS	NVMD						
3	D_LEN			Reserved		STATUS									
4	IPReserved / D_DATA[31:0]														
...															
11															
12	IPBAL / D_DATA[35:32]														
13	IPBAH / D_DATA[39:36]														
14	IPDL / D_DATA[43:40]														
15	IPReserved / D_DATA [47:44]														

Table 177 GET_NVMD_DATA Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the GET_NVMD_DATA Command .

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
2 [31]	Indirect Payload	IP	1 bit	<p>Flag indicating the data payload mode:</p> <p>0b: Direct payload mode. Use DWords [15:4] in the outbound IOMB for the data response.</p> <p>1b: Indirect payload mode. Use DWords [15:14] as the physical address for the data response buffer in host memory.</p> <p>Note: If NVMD = 0001b (Configuration EEPROM device), 0100b (VPD flash memory), 0101b (AAP1 register dump or event log flash memory) and 0110b (IOP register dump or event log flash memory), only IP=1b mode is supported.</p>
2 [23:16]	TWI Device Address	TDA	1 Byte	The address of the TWI device to read from as specified in the GET_NVMD_DATA Command .
2 [15:12]	TWI Bus Number	TBN	4 bits	TWI bus number as specified in the GET_NVMD_DATA Command .
2 [11:8]	TWI Device Page Size	TDPS	4 bits	TWI Device Page Size as specified in the GET_NVMD_DATA Command .
2 [7:4]	TWI Device Address Size	TDAS	4 bits	TWI Device Address Size as specified in the GET_NVMD_DATA Command .
2 [3:0]	NVM Device	NVMD	4 bits	Device to access as specified in the GET_NVMD_DATA Command .
3 [31:24]	Direct Payload Data Length	D_LEN	1 Byte	Direct data payload length in bytes as specified in the GET_NVMD_DATA Command , valid only if IR is set to 0. The maximum length is 48 bytes.
3 [15:0]	Response Status	STATUS	2 Bytes	<p>Status of the operation:</p> <p>0x0000: SUCCESSFUL. The request completed successfully.</p> <p>0x0001: Payload mode and NVMD combination error. Only NVMD=0000b (TWI devices) support both direct payload mode and indirect payload mode. For other devices, only indirect mode is supported.</p> <p>0x0002: Payload length error. For indirect mode, the maximum length is 4 Kbytes. For direct mode, the maximum length is 48 bytes.</p> <p>0x2001: The TWI device returns a NACK.</p> <p>0x2002: The TWI device lost arbitration.</p> <p>0x2021: TWI device operation time out.</p> <p>0x2081: The TWI bus returns a NACK.</p> <p>0x2082: TWI device arbitration failure.</p> <p>0x20FF: TWI bus serial timeout.</p> <p>0x9001: The specified partition number is not in the flash memory configuration.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				0x9002: The specified length is too large. 0x9003: A flash memory programming operation failed. 0x9004: A flash memory device ID does not match the FLASH configuration. 0x9005: A flash memory vendor ID does not match the flash memory configuration. 0x9006: The flash memory sector erase operation timed out. 0x9007: The flash memory sector erase operation completed but with errors. 0x9008: The flash memory device is busy (internal programming or erase state). 0x9009: The flash memory driver does not support this device. 0x900A: The device does not have a CFI interface. 0x900B: The device has more erase blocks than currently supported. 0x900C: The specified partition is read only and the operation could not be completed. 0x900D: The specified partition map type is invalid. 0x900E: The partition map from the initialization string is disabled.
11:4	Reserved for Indirect Payload Mode / Direct Payload Data	IPReserved / D_DATA[31:0]	32 Bytes	If IP=1b, this is reserved. If IP=0b, this is the direct payload data bytes [31:0].
12	Indirect Payload Buffer Address Lower / Direct Payload Data	IPBAL / DATA[35:32]	4 Bytes	If IP=1b, this is the lower 32-bit physical address for the response buffer in host memory. If IP=0b, this is the direct payload data bytes [35:32].
13	Indirect Payload Buffer Address Higher / Direct Payload Data	IPBAH / DATA[39:36]	4 Bytes	If IP=1b, this is the higher 32-bit physical address for the response buffer in host memory. If IP=0b, this is the direct payload data bytes [39:36].
14	Indirect Payload Data Length / Direct Payload Data	IPDL / DATA[43:40]	4 Bytes	If IP=1b, this is the size of the indirect payload mode data. The maximum length is 4 Kbytes. If IP=0b, this is the direct payload data bytes [43:40].

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
15	Reserved for Indirect Payload Mode / Direct Payload Data	IPReserved / D_DATA [47:44]	4 Bytes	If IP=1b, this is reserved. If IP=0b, this is the direct payload data bytes [47:44].

8.29 SET_NVMD_DATA Response

Description

The SPC 8x6G controller sends this response as the response to the [SET_NVMD_DATA Command](#) described in Section 7.28.

Usage

Initiator and target.

Response Format

Table 178 SET_NVMD Response Format

	Byte 3			Byte 2		Byte 1		Byte 0							
0	V	H	R	BC=1		Reserved	OBID	CAT=0x2 OPC=0x024							
1	HTAG														
2	IP	Reserved		TDA		TBN	TDPS	TDAS NVMD							
3	Reserved					STATUS									
4	Reserved														
...															
15															

Table 179 SET_NVMD Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SET_NVMD_DATA Command .
2 [31]	Indirect Payload	IP	1 bit	Flag indicating the data payload mode: 0b: Direct payload mode. Use DWords [15:4] in outbound IOMB for the data response. 1b: Indirect payload mode. Use DWords [15:14] as the physical address for the data response buffer in host memory. Note: If NVMD = 0001b (Configuration EEPROM device) or 0100b (VPD flash memory), only IP=1b mode is supported.
2 [23:16]	TWI Device Address	TDA	1 Byte	The address of the TWI device to read from as specified in the GET_NVMD_DATA Command .
2 [15:12]	TWI Bus Number	TBN	4 bits	TWI bus number as specified in the GET_NVMD_DATA Command .
2 [11:8]	TWI Device Page Size	TDPS	4 bits	TWI Device Page Size as specified in the GET_NVMD_DATA Command .
2 [7:4]	TWI Device Address Size	TDAS	4 bits	TWI Device Address Size as specified in the GET_NVMD_DATA Command .
2 [3:0]	NVM Device	NVMD	4 bits	Device to access as specified in the GET_NVMD_DATA Command .
3 [15:0]	Response Status	STATUS	2 Bytes	Status of the operation: 0x0000: SUCCESSFUL. The request completed successfully. 0x0001: ERROR Payload mode and NVMD combination error. Only NVMD=0000b (TWI devices) support both direct payload mode and indirect payload mode. (Note: if programming the configuration EEPROM space, only indirect mode is supported.) For other devices, only indirect mode is supported. 0x0002: Payload length error. For indirect mode, the maximum length is 4 Kbytes. For direct mode, the maximum length is 48 bytes. 0x0003: Data address offset is not set to 0x0. For flash device and the configuration EEPROM, the DOA should be set to 0x0. 0x0004: Signature error. For setting the configuration EEPROM, the signature should be set to 0xFEDCBA98. 0x0005: TWI address size error. Only 1- and 2-byte addressing sizes are supported. 0x2001: The TWI device returns a NACK. 0x2002: The TWI device lost arbitration. 0x2021: TWI device operation time out.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>0x2081: The TWI bus returns a NACK.</p> <p>0x2082: TWI device arbitration failure.</p> <p>0x20FF: TWI bus serial timeout.</p> <p>0x9001: The specified partition number is not in the flash memory configuration.</p> <p>0x9002: The specified length is too large.</p> <p>0x9003: A flash memory programming operation failed.</p> <p>0x9004: A flash memory device ID does not match the flash memory configuration.</p> <p>0x9005: A flash memory vendor ID does not match the flash memory configuration.</p> <p>0x9006: The flash memory sector erase operation timed out.</p> <p>0x9007: The flash memory sector erase operation completed but with errors.</p> <p>0x9008: The flash memory device is busy (internal programming or erase state).</p> <p>0x9009: The flash memory driver does not support this device.</p> <p>0x900A: The device does not have a CFI interface.</p> <p>0x900B: The device has more erase blocks than currently supported.</p> <p>0x900C: The specified partition is read only and the operation could not be completed.</p> <p>0x900D: The specified partition map type is invalid.</p> <p>0x900E: The partition map from the initialization string is disabled.</p>

8.30 DEVICE_HANDLE_REMOVED Notification

Description

This event is sent by the SPC 8x6G controller as a notification of the removal of the device handle as the result of port operation 0x05: HARD_RESET in the [PORT_CONTROL Command](#) described in Section 7.26.

Usage

Target.

Response Format

Table 180 DEVICE_HANDLE_REMOVED Notification Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02 OPC=0x025		
1	Reserved									
2	DEVICE_ID									
3	Reserved									
...										
15										

Table 181 DEVICE_HANDLE_REMOVED Notification Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1 [3:0]	Port Identifier	PORT_ID	4 bits	The ID for the SPC 8x6G controller's local SAS/SATA port context.
2	Device Identifier	DEVICE_ID	4 Bytes	SSP target device identifier. See Section 3.2 for a detailed description of DEVICE_ID.

8.31 SET_DEVICE_STATE Response

Description

This message is sent by the SPC 8x6G controller as the response to the [SET_DEVICE_STATE Command](#) described in Section 7.29.

Usage

Initiator and target.

Response Format

Table 182 SET_DEVICE_STATE Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1		Reserved	OBID	CAT=0x02	OPC=0x026	
1	HTAG									
2	STATUS									
3	DEVICE_ID									
4	Reserved						PDS	NDS		
5	Reserved									
...										
15										

Table 183 SET_DEVICE_STATE Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SET_DEVICE_STATE Command .
2	Status	STATUS	4 Bytes	<p>Status of the operation.</p> <p>0x00000000: SUCCESS. The device state has been successfully set to New Device State (NDS) specified in the SET_DEVICE_STATE Command. The NDS field indicates the new device state and the PDS field indicates the previous device state before transition.</p> <p>0x00000001: INVALID_STATE. The state change is not permitted for current device state.</p> <p>0x00000007: IO_NO_DEVICE. An I/O request was sent to a non-existent device (invalid DEVICE_ID). Both the NDS and PDS fields are not used.</p>

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
3	Device Identifier	DEVICE_ID	4 Bytes	Device Identifier. See Section 3.2, "Device Handle and DEVICE_ID" for a detailed description of DEVICE_ID.
4 [3:0]	New State	NDS	4 bits	Indicates the state of the device after the state change. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states.
4 [7:4]	Previous State	PDS	4 bits	Indicates the state of the device before the state change. See Table 8 in Section 3.2, "Device Handle and DEVICE_ID" for the definition of the device states.

8.32 GET_DEVICE_STATE Response

Description

This message is sent by the SPC 8x6G controller as the response to the [GET_DEVICE_STATE Command](#) described in Section 7.30.

Usage

Initiator and target.

Response Format

Table 184 GET_DEVICE_STATE Response Format

	Byte 3			Byte 2		Byte 1		Byte 0
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x027
1	HTAG							
2	STATUS							
3	DEVICE_ID							
4	Reserved							DS
5	Reserved							
...								
15								

Table 185 GET_DEVICE_STATE Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the GET_DEVICE_STATE Command .
2	Status	STATUS	4 Bytes	Status of the operation: 0x00000000: SUCCESS. The DS field contains the device state. 0x00000007: IO_NO_DEVICE. An I/O request was sent to a non-existent device (invalid DEVICE_ID).
3	Device Identifier	DEVICE_ID	4 Bytes	Device Identifier. See Section 3.2, " Device Handle and DEVICE_ID " for a detailed description of DEVICE_ID.
4 [3:0]	Device State	DS	4 bits	Indicates the device state of the device: 0x1 : DS_OPERATIONAL 0x2: DS_PORT_IN_RESET 0x3 : DS_IN_RECOVERY 0x4 : DS_IN_ERROR 0x5 : Reserved 0x6 : Reserved 0x7 : DS_NON_OPERATIONAL

8.33 SET_DEVICE_INFO Response

Description

This message is sent by the SPC 8x6G controller as the response to the [SET_DEVICE_INFO Command](#) described in Section 7.31.

Usage

Initiator and target.

Response Format

Table 186 SET_DEVICE_INFO Response Format

	Byte 3				Byte 2		Byte 1		Byte 0										
0	V	H	R	BC=1	Reserved	OBID	CAT=0x02	OPC=0x028											
1	HTAG																		
2	STATUS																		
3	DEVICE_ID																		
4	Reserved										SA	SR	SI						
5	Reserved				A	R	IT Nexus Timeout (ITNT)												
6	Reserved																		
...	Reserved																		
15																			

Table 187 SET_DEVICE_INFO Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SET_DEVICE_INFO Command .
2	Status	STATUS	4 Bytes	Status of the operation: 0x00000000: SUCCESS. 0x00000007: IO_NO_DEVICE. An I/O request was sent to a non-existent device (invalid DEVICE_ID).
3	Device Identifier	DEVICE_ID	4 Bytes	Device Identifier. See Section 3.2, "Device Handle and DEVICE_ID" for a detailed description of DEVICE_ID.
4 [2]	Set AWT	SA	1 bit	SA field value set in SET_DEVICE_INFO Command .
4 [1]	Set Retry Flag	SR	1 bit	SR field value set in SET_DEVICE_INFO Command .
4 [0]	Set ITNT	SI	1 bit	SI field value set in SET_DEVICE_INFO Command .
5 [17]	AWT flag	A	1 bit	Latest value of the priority setting for the Arbitration Wait Time (AWT) for this device: 0b: Default setting (recommended). The actual AWT value is based on how long an OPEN frame has been waiting for a connection request to be accepted. It starts at 0. As specified in the SAS specification, from 0 to 32768 µs, the AWT value is incremented every µs. From 32768 µs and on, the AWT value is incremented every ms. 1b: Increase priority. The actual AWT value starts at 32768 µs when the A flag is set to '1'. The AWT value is incremented every ms from that point on.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
5 [16]	Retry flag	R	1 bit	Latest value of the Transport Layer Retry (TLR) flag per the SAS 1.1 and SAS 2.0 specifications: 1b: Enable TLR 0b: Disable TLR
5 [15:0]	ITNT	ITNT	2 Bytes	Valid if the SI field is set to 01b. Latest value in milliseconds of the time unit that is used by the SPC 8x6G to determine the nexus timeout condition.

8.34 SAS_RE_INITIALIZATION Response

Description

This message is sent by the SPC 8x6G controller as the response to the [SAS_RE_INITIALIZATION Command](#) described in Section [7.32](#).

Usage

Initiator and target.

Response Format

Table 188 SAS_RE_INITIALIZATION Response Format

	Byte 3				Byte 2		Byte 1		Byte 0			
0	V	H	R	BC=1	Reserved		OBID	CAT=0x02	OPC=0x029			
1	HTAG											
2	STATUS											
3	Reser ved	S M P O R R C	S O R R D	S O R N D	S S A H O L T	Reserved						
4	Reserved					Reserved	MAX_PORTS					
5	OPEN_REJECT_RETRIES_CMD					OPEN_REJECT_RETRIES_DATA						
6	Reserved					SATA_HOL_TMO						
7	Reserved											

	Byte 3	Byte 2	Byte 1	Byte 0
...				
15				

Table 189 SAS_RE_INITIALIZATION Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context as specified in the SAS_RE_INITIALIZATION Command .
2	Status	STATUS	4 Bytes	<p>Status of the operation:</p> <p>0x00000000: SUCCESS.</p> <p>0x00000001: CONTROLLER_NOT_IDLE. The SPC 8x6G is not idle to perform the SAS reconfiguation operation. Quiesce the SPC 8x6G SAS/SATA operation before issuing a SAS_RE_INITIALIZATION Command.</p> <p>0x00000002: INVALID_CONFIG_PARAM. One or more of the command parameters are invalid (out of range). For example, MAX_PORTS is set to a value that is not supported by the controller.</p>
3 [31]	Reserved		1 bit	Reserved
3 [30]	Reserved		1 bit	Reserved
3 [29]	Set MAX PORTS	SMPORT	1 bit	Set MAX PORTS flag set in the SAS_RE_INITIALIZATION Command .
3 [28]	Set OPEN REJECT RETRIES CMD	SORRC	1 bit	Set OPEN REJECT RETRIES Command flag set in the SAS_RE_INITIALIZATION Command .
3 [27]	Set OPEN REJECT RETRIES DATA	SORRD	1 bit	Set OPEN REJECT RETRIES DATA flag set in the SAS_RE_INITIALIZATION Command .
3 [26]	Set OPEN REJECT NO DESTINATION OPTION	SORNDO	1 bit	Set OPEN REJECT NO DESTINATION OPTION flag set in the SAS_RE_INITIALIZATION Command .
3 [25]	Set SATA HOL TMO	SSAHOLT	1 bit	Set SATA HOL TMO flag set in the SAS_RE_INITIALIZATION Command .
3 [24:0]	Reserved			Reserved
4 [31:16]	Reserved		2 Bytes	Reserved
4 [15:8]	Reserved		2 Bytes	Reserved
4 [7:0]	Maximum SAS Ports	MAX_PORTS	2 Bytes	Latest value of the number of SAS ports supported.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
5 [31:16]	OPEN REJECT (RETRY) Command Phase	OPEN_REJ_ECT_RETRI_ES_CMD	2 Bytes	Latest value of the number of retries supported for an OPEN REJECT (RETRY) during the command phase.
5 [15:0]	OPEN REJECT (RETRY) Data Phase	OPEN_REJ_ECT_RETRI_ES_DATA	2 Bytes	Latest value of the number of retries supported for an OPEN REJECT (RETRY) during the data phase.
6 [15:0]	SATA Head-Of-Line Blocking Detection Timeout	SATA_HOL_TMO	2 Bytes	Latest value of the timeout delay used to detect SATA Head-Of-Line Blocking. The delay is in 100 millisecond units.
6 [31:16]	Reserved		2 Bytes	Reserved

8.35 SGPIO Response

Description

This message is sent by the SPC 8x6G controller as the response to the SGPIO_REGISTER Command described in Section 7.33.

Usage

Initiator.

Response Format

Table 190 SGPIO Response Format

	Byte 3			Byte 2		Byte 1		Byte 0				
0	V	H	R	BC=1			Reserved	OBID	CAT=0x02	OPC=0x082E		
1	HTAG											
2	Reserved1			SgpioFunctionResult		SgpioFunction		SgpioSMPFrameType				
3	Payload											
4												
5												
...												
15												

Table 191 SGPIO Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [7:0]	SgpioSMPFrameType	SSFT	1 byte	Set the SMP Frame Type field to 0x41. Refer to SFF-8485 Specification for Serial GPIO (GPIO) Bus.
2 [15:8]	SgpioFunction	SF	1 byte	This field needs to be set as follows: 0x02: READ GPIO REGISTER 0x82: WRITE GPIO REGISTER Refer to SFF-8485 Specification for Serial GPIO (GPIO) Bus.
2 [23:16]	SgpioFunctionResult	SFR	1 byte	GPIO Function results : 0x0: GPIO_COMMAND_SUCCESS 0x1:GPIO_CMD_ERROR_WRONG_FRAME_TYPE 0x2:GPIO_CMD_ERROR_WRONG_REG_TYPE 0x3:GPIO_CMD_ERROR_WRONG_REG_INDEX 0x4:GPIO_CMD_ERROR_WRONG_REG_COUNT 0x5:GPIO_CMD_ERROR_WRONG_FRAME_REG_TYPE 0x6:GPIO_CMD_ERROR_WRONG_FUNCTION 0x19:GPIO_CMD_ERROR_WRONG_FRAME_TYPE_REG_INDEX 0x81:GPIO_CMD_ERROR_WRONG_FRAME_TYPE_REG_CNT 0x1A:GPIO_CMD_ERROR_WRONG_REG_TYPE_REG_INDEX 0x82:GPIO_CMD_ERROR_WRONG_REG_TYPE_REG_COUNT 0x83:GPIO_CMD_ERROR_WRONG_REG_INDEX_REG_COUNT 0x1D:GPIO_CMD_ERROR_WRONG_FRAME_REG_TYPE_REG_INDEX 0x9D:GPIO_CMD_ERROR_WRONG_ALL_HEADER_PARAMS
2 [31:24]	Reserved 1	R1	1 byte	Refer to SFF-8485 Specification for Serial GPIO (GPIO) Bus.

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
3-15	Payload	Payload	52 bytes	The payload is 52 bytes in length. It is based on the SGPIO read/write command.

8.36 PCIE_DIAG_EXECUTE Response

Description

The SPC 8x6G controller sends this response to the PCIE_DIAG_EXECUTE Command described in Section 7.34.

Usage

Initiator and target.

Table 192 PCIE_DIAG_EXECUTE Response Format

	Byte 3			Byte 2		Byte 1		Byte 0		
0	V	H	R	BC=1			Reserved	OBID	CAT=0x02	OPC=0x82F
1	HTAG									
2	Reserved				CMD_TY PE		CMD_DESC	Reserved		
3	STATUS									
4	Reserved									
...										
15										

Table 193 PCIE_DIAG_EXECUTE Response Fields

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
1	Host Tag	HTAG	4 Bytes	Tag or context for this operation.
2 [15:13]	Command Type	CMD_TYPE	3 bits	Command type set in PCIE_DIAG_EXECUTE Command.
2 [12:8]	Command Descriptor	CMD_DESC	5 bits	Command descriptor set in PCIE_DIAG_EXECUTE Command.
3 [31:0]	Status	STATUS	32 bits	Status of the operation:

DWord [Bit(s)]	Field Name	Field Symbol	Length	Description
				<p>0x0: SUCCESS Data was successfully read/written to the host. The host must verify the contents of the buffer.</p> <p>0x1: INVALID_COMMAND Operation aborted due to command fields being formatted incorrectly.</p> <p>0x2 : INTERNAL_FAILURE Operation aborted due to SPC 8x6G internal failure.</p>

9 PCIe Registers

9.1 Programming Interface

The SPC 8x6G controller is a single-function device and responds to function 0 for configuration cycles.

All PCIe registers are byte accessible.

Writes to reserved registers have unspecified consequences including, but not limited to, PCIe unit reset. Reads of reserved registers or address space return undefined data.

The PCIe registers in this chapter are listed in the order of the configuration address.

Please see the PCI Express 2.0 Base Specification for the detailed description of these registers.

9.2 PCIe Configuration Space

9.2.1 PCIe Configuration Register Map

Table 194 PCIe Configuration Register Map

PCI Configuration Register				Configuration Address
Byte 3	Byte 2	Byte 1	Byte 0	
DEVICE_ID		VENDID		0x00
CFGSTAT		CFGCMD		0x04
CLSCODE			REVID	0x08
BIST	HDRTYPE	LATTIM <i>Master Latency Timer Register</i>	CLSIZE	0x0C
MEMBASE-I LOWER				0x10
MEMBASE-I UPPER				0x14
MEMBASE-II LOWER				0x18
MEMBASE-II UPPER				0x1C
MEMBASE-III				0x20
MEMBASE-IV				0x24
Reserved				0x28
SVID				0x2C
ROMBASE				0x30

PCI Configuration Register				Configuration Address		
Reserved		CAP_PTR		0x34		
Reserved			0x38			
MAXLAT	MINGNT	INTPIN	INTLINE	0x3C		
PMC	PM_NEXT_CAP	PM_CAP_ID	0x40			
PMCSR			0x44			
Reserved			0x48..0x4F			
MC	MSI_NEXT_CAP	MSI_CAP_ID	0x50			
Message Address Lower			0x54			
Message Address Upper			0x58			
Reserved	Message Data		0x5C			
PCIE_CAP	PCIE_NEXT_CAP	PCIE_CAP_ID	0x70			
DEVICE_CAP			0x74			
DEVICE_STAT	DEVICE_CTRL		0x78			
LINK_CAP			0x7C			
LINK_STAT	LINK_CTRL		0x80			
MSIX_CAP			0xAC			
TBL_OFFSET			0xB0			
PBA_OFFSET			0xB4			
PCIE_CAP_HD			0x100			
UE_STAT			0x104			
UE_MASK			0x108			
UE_SEV			0x10C			
CE_STAT			0x110			
CE_MASK			0x114			
ADV_ERR_CTRL			0x118			
HD_LOG_DW0			0x11C			
HD_LOG_DW1			0x120			
HD_LOG_DW2			0x124			
HD_LOG_DW3			0x128			

9.2.2 PCIe MSI-X Vector Table Address Map

Table 195 PCIe MSI-X Vector Table Address Map

MSI-X Table Structure				MSI-X Entry Number
DWord 3	DWord 2	DWord 1	DWord 0	
Vector Control	Message Data	Message Upper Address	Message Address	0
Vector Control	Message Data	Message Upper Address	Message Address	1
Vector Control	Message Data	Message Upper Address	Message Address	2
Vector Control	Message Data	Message Upper Address	Message Address	3

9.2.3 Register Access and Reset Values

A PCIe Link Reset has the same effect as a power-management reset. A soft reset is a subset of PCIe Link Reset.

Table 196 Register Access and Reset Values

PCIe Register Name	Mnemonic	Config Address (Hex)	Memory Offset (Hex)	Size (Bytes)	Access (Read/Write)	Reset State (Hex)
Vendor Identification Configuration	VENDID	0x00		2	RO	0x11F8
Device Identification	DEVICE_ID	0x02		2	RO	0x8001
Configuration Command	CFGCMD	0x04		2	RW	0x0000
Configuration Status	CFGSTAT	0x06		2	RW1C ¹	0x0010
Revision	REVID	0x08		1	RO	0x05
Class Code	CLSCODE	0x09		3	RO	0x010700
Cache Line Size	CLSIZE	0x0C		1	RW	0x00
Master Latency Timer	LATTIM	0x0D		1	RO	0x00
Header Type	HDRTYPE	0x0E		1	RO	0x00
Built-In Self Test	BIST	0x0F		1	RO	0x00
Memory Base Address I Lower	MEMBASEL-I	0x10		4	RW	0x00000004
Memory Base Address I Upper	MEMBASEU-I	0x14		4	RW	0x00000000
Memory Base Address II Lower	MEMBASEL-II	0x18		4	RW	0x00000004

¹RW1C is a designation for: Read, Write ‘1’ to Clear

PCIe Register Name	Mnemonic	Config Address (Hex)	Memory Offset (Hex)	Size (Bytes)	Access (Read/Write)	Reset State (Hex)
Memory Base Address II Upper	MEMBASEU-II	0x1C		4	RW	0x00000000
Memory Base Address III	MEMBASE-III	0x20		4	RW	0x00000000
Memory Base Address IV	MEMBASE-IV	0x24		4	RW	0x00000000
Subsystem ID and Subsystem Vendor ID	SVID	0x2C		4	RO	0x00000000
ROM Base Address ²	ROMBASE	0x30		4	RO	0x00000001
Configuration Capabilities Pointer	CAP_PTR	0x34		1	RO	0x40
Interrupt Line	INTLINE	0x3C		1	RW	0xFF
Interrupt Pin	INTPIN	0x3D		1	RO	0x01
Minimum Grant	MINGNT	0x3E		1	RO	0x00
Maximum Latency	MAXLAT	0x3F		1	RO	0x00
Power Management Capabilities Identifier	PM_CAP_ID	0x40		1	RO	0x01
Power Management Next Capabilities	PM_NEXT_CAP	0x41		1	RO	0x50
Power Management Capabilities	PMC	0x42		2	RO	0x0BC3
Power Management Control and Status	PMCSR	0x44		4	RW1C	0x00000008
Message Signaled Interrupts Capabilities Identifier	MSI_CAP_ID	0x50		1	RO	0x05
Message Signaled Interrupts Next Capability	MSI_NEXT_CAP	0x51		1	RO	0x70
Message Control Register	MC	0x52		2	RW	0x008A
Message Address Lower	MLA	0x54		4	RW	0x00000000
Message Upper Address	MUA	0x58		4	RW	0x00000000
Message Data	MESG_DATA	0x5C		2	RW	0x0000
Capabilities Identifier	PCIE_CAP_ID	0x70		1	RO	0x10
Next Capabilities	PCIE_NEXT_CAP	0x71		1	RO	0xAC
Capabilities	PCIE_CAP	0x72		2	RO	0x0002
Device Capabilities	DEVICE_CAP	0x74		4	RO	0x00008701
Device Control	DEVICE_CTRL	0x78		2	RW	0x2810
Device Status	DEVICE_STAT	0x7A		2	RW1C	0x0000
Link Capabilities	LINK_CAP	0x7C		4	RO	0x00033C82

² For details see Section 3.23 Expansion ROM Support.

PCIe Register Name	Mnemonic	Config Address (Hex)	Memory Offset (Hex)	Size (Bytes)	Access (Read/Write)	Reset State (Hex)
Link Control	LINK_CTRL	0x80		2	RW	0x0000
Link Status	LINK_STAT	0x82		2	RO	Varies
MSI-X Capability	MSIX_CAP	0xAC		4	RW	0x000F0011
Table Offset	TBL_OFFSET	0xB0		4	RO	0x00002000
PBA Offset	PBA_OFFSET	0xB4		4	RO	0x00004000
Enhanced Capability Header	PCIE_CAP_HD	0x100		4	RO	0x00010001
Uncorrectable Error Status	UE_STAT	0x104		4	RW1CS ³	0x00000000
Uncorrectable Error Mask	UE_MASK	0x108		4	RWS	0x00000000
Uncorrectable Error Severity	UE_SEV	0x10C		4	RWS	0x00062030
Correctable Error Status	CE_STAT	0x110		4	RW1CS	0x00000000
Correctable Error Mask	CE_MASK	0x114		4	RWS ⁴	0x00002000
Advanced Error Capabilities and Control	ADV_ERR_CTRL	0x118		4	RWS	0x000000A0
Header Log	HD_LOG_DW0	0x11C		4	ROS	0x00000000
Header Log	HD_LOG_DW1	0x120		4	ROS ⁵	0x00000000
Header Log	HD_LOG_DW2	0x124		4	ROS	0x00000000
Header Log	HD_LOG_DW3	0x128	0x528	4	ROS	0x00000000

³RW1CS is a designation for: Read, Write '1' to Clear, Sticky (not initialized or modified by reset).

⁴RWS is a designation for: Read, Write, Sticky (not initialized or modified by reset).

⁵ROS is a designation for: Read-Only, Sticky (not initialized or modified by reset).

9.2.4 Vendor Identification Configuration Register

Mnemonic: VENDID

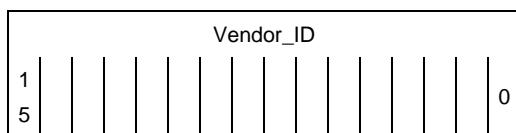
Configuration Address: 0x00

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x11F8

Description: Specifies PMC-Sierra's vendor identification



9.2.5 Device Identification Register

Mnemonic: DEVICE_ID

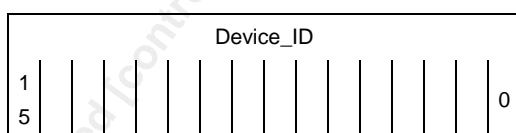
Configuration Address: 0x02

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x8001

Description: Specifies the vendor's device identification number.



9.2.6 Configuration Command Register

Mnemonic:	CFGCMD
Configuration Address:	0x04
Memory Offset:	0x304
Size:	16 Bits
Access Privilege:	PCI Configuration (Read-Write), PCI Memory (Read-Write)
Reset State:	0x0000
Description:	Provides coarse control over the chip's ability to generate and respond to PCIe.

Res	INT_DIS	FBB_EN	SERR_EN	W_CC	PER_EN	VG_A	M_WI	SP_C	MS_T	ME_M	IO_A
1 5	10	9	8	7	6	5	4	3	2	1	0

Table 197 Configuration Command Register

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[15:11]	Res	Reserved	Reserved	0x0	RO
10	INT_DIS	Interrupt Disable	Controls the ability of a PCIe function to generate Assert_INTx messages. When set, the function is prevented from generating Assert_INTx messages. If an Assert_INTx message was already sent by the function, a Deassert_INTx message is sent when this bit is set.	0x0	RW
9	FBB_EN	Fast Back-To-Back Transactions Enable	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
8	SERR_EN	SERR Enable	This bit when set, enables reporting of non-fatal and fatal PCIe errors detected by the function. The errors are reported if enabled either through this bit or through the PCIe specific bits in the Device Control Register .	0x0	RW

Bit(s)	Symbol	Name	Description	Reset State	Attributes
7	WCC	IDSEL Stepping/Wait Cycle Control	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
6	PER_EN	Parity Error Response Enable	0: Disable parity error reporting 1: Enable parity error reporting If parity error reporting is enabled, the poisoned packets received are reported in the Configuration Status Register , where the MDPE bit is set.	0x0	RW
5	VGA	VGA Palette Snoop	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
4	MWI	Memory Write and Invalidate	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
3	SPC	Special Cycle Enable	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
2	MST	Bus Master Enable	Clearing this bit prevents a PCIe agent from issuing any memory requests. Note that since MSI interrupt messages are in-band memory writes, clearing this bit disables MSI interrupt Messages as well. However, split completions can be initiated regardless of the state of this bit.	0x0	RW
1	MEM	Slave Memory Access Enable	When set, enables responding to memory space accesses.	0x0	RW
0	IOA	Slave I/O Access Enable	When set, enables responding to I/O space accesses.	0x0	RW

9.2.7 Configuration Status Register

Mnemonic:	CFGSTAT
Configuration Address:	0x06
Size:	16 Bits
Access Privilege:	PCI Configuration (Read-Write), PCI Memory (Read-Write)
Reset State:	0x0010
Description:	Records status information for PCIe link events. Reads to this register behave normally. Writes are slightly different: bits can be reset, but not set. A bit is reset whenever the register is written and the data in the corresponding bit location is a 1. For example, to clear bit 14 without affecting any other bits, write the value 0100_0000_0000_0000b to the register.

Also see the “Error Information” section.



Table 198 Configuration Status Register

Bit(s)	Symbol	Name	Description	Reset State	Attributes
15	DPE	Detected Parity Error	This bit is set whenever a poisoned TLP is received, regardless of the state of the Parity Error Response Enable (PER_EN) bit in the Configuration Command Register .	0x0	RW1C (Read, write logic 1 to clear)
14	SSE	Signaled System Error	Set when a device sends an ERR_FATAL or ERR_NONFATAL message, and the SERR Enable bit in the Command Register is 1. ERR_NONFATAL messages can be function-specific. In this case, only the function generating the message has this bit set if appropriate.	0x0	RW1C
13	RMA	Received Master Abort	Set when a requestor receives a completion with Unsupported Request Completion Status.	0x0	RW1C

Bit(s)	Symbol	Name	Description	Reset State	Attributes
12	RTA	Received Target Abort	Set when a requestor receives a completion with Completer Abort Completion Status.	0x0	RW1C
11	STA	Signaled Target Abort	Set when a device completes a request using a Completer Abort Request Status.	0x0	RW1C
[10:9]	DST	DEVSEL Timing	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
8	MDPE	Master Data Parity Error	<p>Set by a requestor if its Parity Error Response Enable (PER_EN) bit (in the Configuration Command Register) is set and it 1) receives a completion marked poisoned or 2) poisons a write request.</p> <p>This bit is not set if the PER_EN bit is cleared.</p>	0x0	RW1C
7	FBB	Fast Back-To-Back Transactions Capable	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
6	Res	Reserved	Reserved.	0x0	RO
5	66M	66 MHz Capable	Does not apply to PCIe. This is a read-only field that is always zero.	0x0	RO
4	CPL	Capabilities List	Indicates the presence of an extended capability list item. The pointer to the first item in the capabilities link list is at offset 0x34.	0x1	RO
3.	INTSTAT	Interrupt Status	Indicates that an INTx interrupt message is pending internally to the device. If MSI or EMI is enabled, this bit is not set. Clearing the interrupt source register bit clears this bit.	0x0	RO
[2:0]	Res	Reserved	Reserved	0x0	RO

9.2.8 Revision Register

Mnemonic: REVID

Configuration Address: 0x08

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: Dependent upon device revision level (0x4 to indicate 1.0 initially)

Description: This revision level is made up of a Major and a Minor revision field. The value, when read from this register, is interpreted as “Major.Minor”. Increasing values in either Major or Minor fields indicate later revisions.

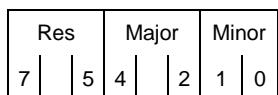
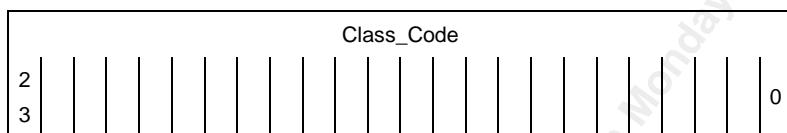


Table 199 Revision Register

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[7:5]	Res	Reserved	Reserved	0x0	RO
[4:2]	Major	Major Revision Field	Represents a major change to the silicon, such as a full mask change.	0x1	RO
[1:0]	Minor	Minor Revision Field	Represents a minor change to the silicon, such as a metal layer only change.	0x0 (Rev B) 0x1 (Rev C)	RO

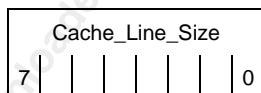
9.2.9 Class Code Register

Mnemonic: CLSCODE
 Configuration Address: 0x09
 Size: 24 Bits
 Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)
 Reset State: 0x010700
 Description:



9.2.10 Cache Line Size Register

Mnemonic: CLSIZE
 Configuration Address: 0x0C
 Memory Offset: 0x30C
 Size: 8 Bits
 Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)
 Reset State: 0x00
 Description: The Cache Line Size Register is set by the system firmware and the operating system to system cache line size. However, this register has no impact on any PCIe device functionality.



9.2.11 Master Latency Timer Register

Mnemonic: LATTIM

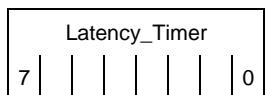
Configuration Address: 0x0D

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00

Description: The master latency timer does not apply to PCIe. This register is always read as zero.



9.2.12 Header Type Register

Mnemonic: HDRTYPE

Configuration Address: 0x0E

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00

Description: Indicates that the chip is a multi-function device using header type.

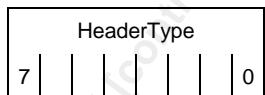


Table 200 Header Type Register

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[7:0]	Header_Type	Header Type	The only valid value is 00h, which indicates that this PCIe endpoint device has a single function, using header type 0.	0x00	RO

9.2.13 Built-In Self Test Register

Mnemonic: BIST

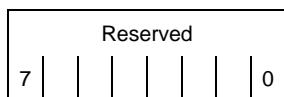
Configuration Address: 0x0F

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00

Description: Built-In Self Test (BIST) Register is not supported. This register is always read as 0x00.



9.2.14 Memory Base Address I Register

Mnemonic:	MEMBASE-I
Configuration Address:	Lower MEMBASE 0x10
	Upper MEMBASE 0x14
Size:	Lower MEMBASE 32 Bits
	Upper MEMBASE 32 Bits
Access Privilege:	PCI Configuration (Read-Write)
Reset State:	Lower MEMBASE 0x00000004
	Upper MEMBASE 0x00000000
Description:	

Maps the register set into the system's memory address space, {Upper MEMBASE, Lower MEMBASE}.

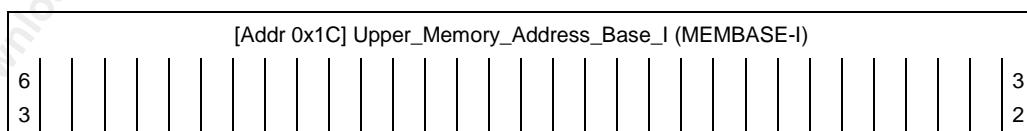
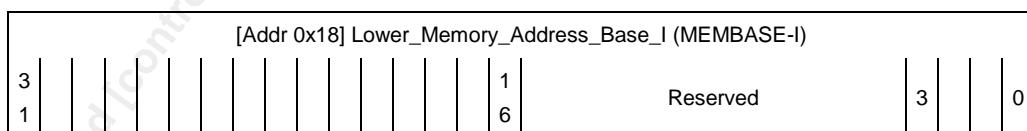
Only the upper 48 bits, bits [63:16], are used for address decoding

Bits [15:4] are read-only and always read as 0

Bit [3] is read as 0 to indicate that the registers are not prefetchable

Bits [2:1] are read as 0b10 to indicate that the registers can be located anywhere in the 64-bit memory address space

Bit [0] is read as 0 to indicate that this register specifies memory space



9.2.15 Memory Base Address II Register

Mnemonic:	MEMBASE-II
Configuration Address:	Lower MEMBASE 0x18
	Upper MEMBASE 0x1C
Size:	Lower MEMBASE 32 Bits
	Upper MEMBASE 32 Bits
Access Privilege:	PCI Configuration (Read-Write), PCI Memory (Read-Write)
Reset State:	Lower MEMBASE 0x00000004
	Upper MEMBASE 0x00000000
Description:	

Maps the register set into the system's memory address space, {Upper MEMBASE, Lower MEMBASE}.

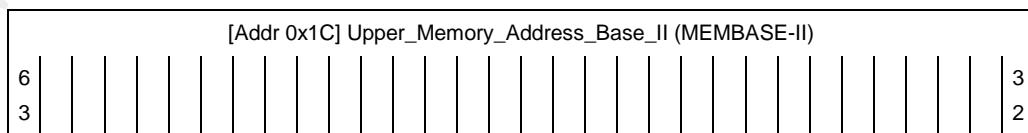
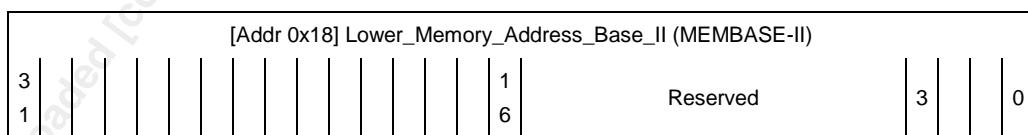
Only the upper 48 bits, bits [63:16], are used for address decoding

Bits [15:4] are read-only and always read as 0

Bit [3] is read as 0 to indicate that the registers are not prefetchable

Bits [2:1] are read as 0b10 to indicate that the registers can be located anywhere in the 64-bit memory address space

Bit [0] is read as 0 to indicate that this register specifies memory space



9.2.16 Memory Base Address III Register

Mnemonic: MEMBASE-III

Configuration Address: MEMBASE 0x20

Size: MEMBASE 32 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: MEMBASE 0x00000000

Description: Maps the register set into the system's memory address space

Maps the register set into the system's memory address space

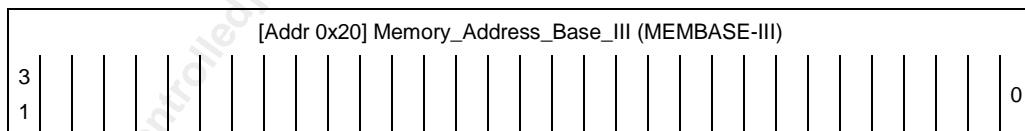
Only the upper 16 bits, bits [31:16], are used for address decoding

Bits [15:4] are read-only and always read as 0

Bit [3] is read as 0 to indicate that the registers are not prefetchable

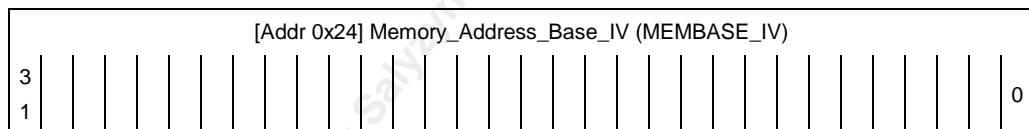
Bits [2:1] are read as 0b00 to indicate that the registers can be located anywhere in the 32-bit memory address space

Bit [0] is read as 0 to indicate that this register specifies memory space



9.2.17 Memory Base Address IV Register

Mnemonic:	MEMBASE-IV
Configuration Address:	MEMBASE 0x24
Size:	MEMBASE 32 Bits
Access Privilege:	PCI Configuration (Read-Write), PCI Memory (Read-Write)
Reset State:	MEMBASE 0x00000000
Description:	<p>Maps the register set into the system's memory address space</p> <p>Only the upper 16 bits, bits [31:16], are used for address decoding</p> <p>Bits [15:4] are read-only and always read as 0</p> <p>Bit [3] is read as 0 to indicate that the registers are not prefetchable</p> <p>Bits [2:1] are read as 0b00 to indicate that the registers can be located anywhere in the 32-bit memory address space</p> <p>Bit [0] is read as 0 to indicate that this register specifies memory space</p>



9.2.18 Subsystem ID and Subsystem Vendor ID Register

Mnemonic: SVID

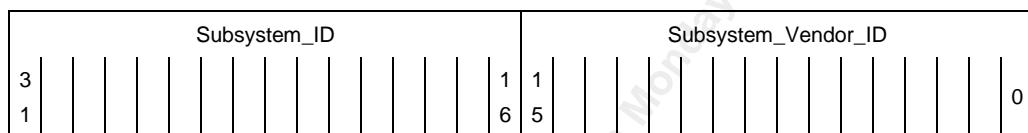
Configuration Address: 0x2C

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00000000

Description: Specifies a subsystem ID and subsystem Vendor ID to a system driver.



9.2.19 ROM Base Address Register

Mnemonic: ROMBASE

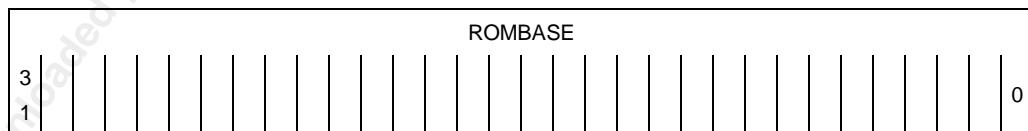
Configuration Address: 0x30

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

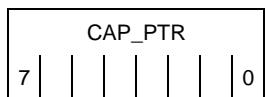
Reset State: 0x00000000

Description: This register is assigned by the system. The value of this register is used by the system to access the expansion ROM space.



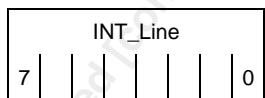
9.2.20 Configuration Capabilities Pointer Register

Mnemonic: CAP_PTR
Configuration Address: 0x34
Size: 8 Bits
Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)
Reset State: 0x40
Description: Provides an offset to the first item in the capabilities linked list.



9.2.21 Interrupt Line Register

Mnemonic: INTLINE
Configuration Address: 0x3C
Size: 8 Bits
Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)
Reset State: 0xFF
Description: This register provides the interrupt line designation assigned by the system. The value of this register is used by device drivers and the operating system. The device itself does not use this register.



9.2.22 Interrupt Pin Register

Mnemonic: INTPIN

Configuration Address: 0x3D

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x01

A value of 0x01 means that INTA_L is used by Function_0.

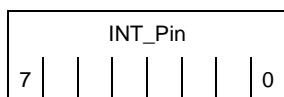
0x01: INTA_L is used by Function 0

Description: 0x02: INTB_L is used by Function 1

0x03: INTC_L is used by Function 2

0x04: INTD_L is used by Function 3

These values are fixed and determined by the function number.



9.2.23 Minimum Grant Register

Mnemonic: MINGNT

Configuration Address: 0x3E

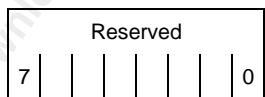
Memory Offset: 0x33E

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00

Description: This register does not apply to PCIe. It is read only and always read as zero.



9.2.24 Maximum Latency Register

Mnemonic: MAXLAT

Configuration Address: 0x3F

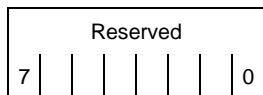
Memory Offset: 0x33F

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00

Description: This register does not apply to PCIe. It is always read as 0x00.



9.2.25 Power Management Capabilities Identifier Register

Mnemonic: PM_CAP_ID

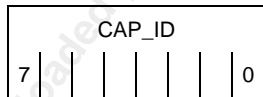
Configuration Address: 0x40

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x01

Description: Identifies the Capabilities Structure. This register contains a value of 0x01 indicating that this is the power management capabilities data structure.



9.2.26 Power Management Next Capabilities Register

Mnemonic: PM_NEXT_CAP

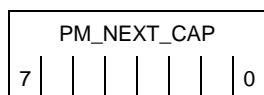
Configuration Address: 0x41

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x50

Description: Pointer to the next item in the capabilities list



9.2.27 Power Management Capabilities Register

Mnemonic: PMC

Configuration Address: 0x42

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x0BC3

Description: Indicates the Power Management Capabilities. Only Power states D0 and D3 are supported.

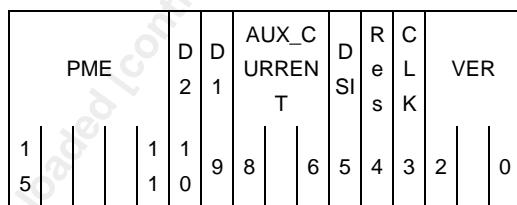


Table 201 Power Management Capabilities Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[15:11]	PME	PME Support	PME generation from D0 is supported.	0x01	RO
10	D2	D2 Support	D2 state is not supported.	0x0	RO
9	D1	D1 Support	D1 state is supported.	0x1	RO
[8:6]	AUX_CU_RRENT	AUX Current	AUX Current Requirement	0x7	RO
5	DSI	Device Specific Initialization	Special initialization is not required.	0x0	RO
4	Res	Reserved	Reserved	0x0	RO
3	CLK	PME Clock	Does not apply to PCIe.	0x0	RO
[2:0]	VER	Version	Power Management Specification Version.	0x3	RO

9.2.28 Power Management Control and Status Register

Mnemonic: PMCSR

Configuration Address: 0x44

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x00000008

Description: Indicates the controls of the Power Management State and the Power Management Status.

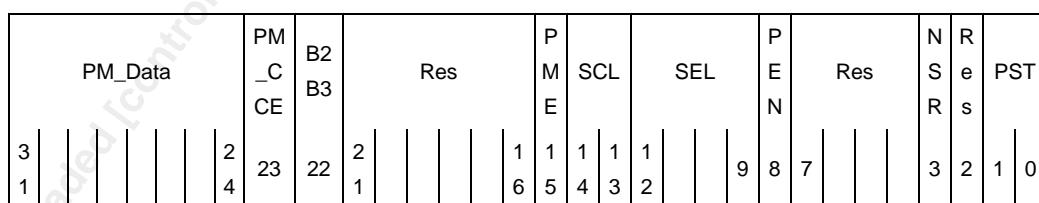


Table 202 Power Management Control and Status Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:24]	PM_Data	Data	Read as 0 because the data register is not supported.	0x0	RO
23	PM_CCE	Bus Power/Clock Control Enable	Does not apply to PCIe. Read as 0.	0x0	RO
22	B2B3	B2/B3 Support	Does not apply to PCIe. Read as 0.	0x0	RO
[21:16]	Res	Reserved	Read as 0.	0x0	RO
15	PME	PME Status	Indicates if a previously enabled PME event occurred or not.	0x0	RW1C
[14:13]	SCL	Data Scale	Read as 0 because the data register is not supported.	0x0	RO
[12:9]	SEL	Data Select	Read as 0 because the data register is not supported.	0x0	RO
8	PEN	PME Enable	PME generation enable.	0x0	RWS
[7:4]	Res	Reserved	Read as 0.	0x0	RO
[3]	NO_SFT_RST	NO_SFT_RS_T	No Soft reset	0x1	RO
[2]	Res	Reserved	Read as 0.	0x0	RO
[1:0]	PST	Power State	<p>Writing a 0x0 to this field enables the D0 power state.</p> <p>No soft reset is performed when going from state D3 to D0. A soft reset can be initiated explicitly by following the procedure described in Section 11.4.2, Soft Reset Recovery, if required.</p> <p>Writing 0x3 to this field enables the D3 power state and the chip performs the following:</p> <ul style="list-style-type: none"> The Interrupt Enable Register is cleared. In the Configuration Command Register, the following bits are cleared: <ul style="list-style-type: none"> SERR Enable Parity Error Response Enable Bus Master Enable PCIe Slave Memory Access Enable PCIe Slave I/O Access Enable 	0x0	RW

9.2.29 Message Signaled Interrupts Capabilities Identifier Register

Mnemonic: MSI_CAP_ID

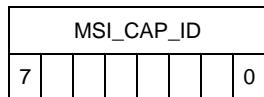
Configuration Address: 0x50

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x05

Description: Identifies the Capabilities Structure. This register contains a value of 0x05 indicating that this is the Message Signaled Interrupt structure.



9.2.30 Message Signaled Interrupts Next Capability Register

Mnemonic: MSI_NEXT_CAP

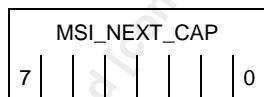
Configuration Address: 0x51

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x70

Description: Pointer to the next item in the capabilities list.



9.2.31 Message Control Register

Mnemonic: MC

Configuration Address: 0x52

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x008A

Description: Provides software control over Message Signaled Interrupts (MSIs).

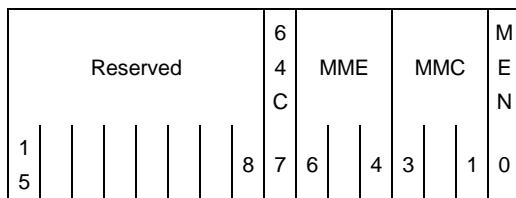


Table 203 Message Control Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[15:8]	Reserved	Reserved	Read as 0.	0x0	RO
7	64C	64 Bit Address Capable	Read as 1 to indicate that the chip is capable of generating a 64-bit message address.	0x1	RO
[6:4]	MME	Multiple Message Enable	This field indicates the number of allocated messages, which may be less than or equal to the MMC. MME# Of Messages Supported 000: 1 001: 2	0x0	RW
[3:1]	MMC	Multiple Message Capable	System software reads this field to determine the number of requested messages. This field indicates the number of messages the device can generate. A value of 0 in this read-only field indicates that the device is capable of generating at most 1 different interrupt messages.	0x5	RO

Bit(s)	Symbol	Name	Description	Reset State	Attributes
0	MEN	MSI Enable	If set to 1, MSIs are enabled and the INT _x _L pin (x is A, B, C) and EMI interrupt mechanisms are not used. If cleared to 0, the MSIs are not used to request service.	0x0	RW

9.2.32 Message Lower Address Register

Mnemonic: MLA

Configuration Address: 0x54

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x00000000

Description: Indicates the system-specified, DWord-aligned lower 32-bit message address.

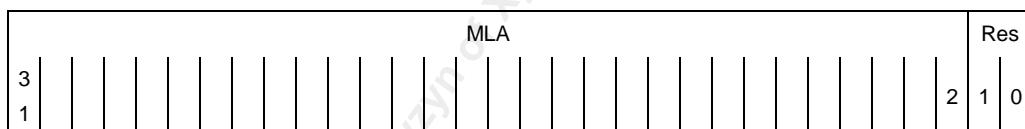


Table 204 Message Address Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:2]	MLA	Message Lower Address	Only valid if the MEN (MSI Enable) bit in the Message Control Register is set to 1.	0x0	RW
[1:0]	Res	Reserved	Reserved. Read as 0.	0x0	RO

9.2.33 Message Upper Address Register

Mnemonic: MUA

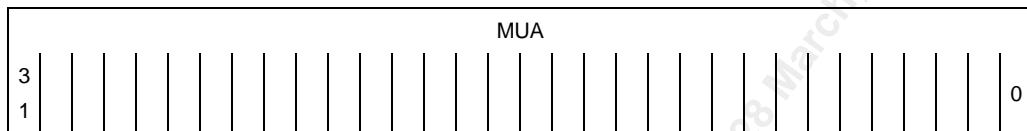
Configuration Address: 0x58

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x00000000

Description: Indicates the system-specified upper 32-bit message address.
Optional, used if '64C = 1'.
If '64C = 0', then the Upper 32-bit Address register functions as the MSI Data Register and register address 0x5C is not used.



9.2.34 Message Data Register

Mnemonic: MESG_DATA

Configuration Address: 0x5C

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x0000

Description: Indicates system specified message. Optional, used if '64C = 0'.

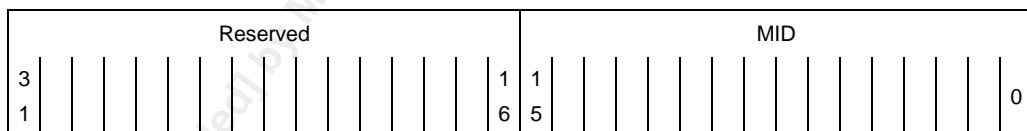


Table 205 Message Data Register

Bit(s)	Symbol	Name	Description
[31:16]	Reserved	Reserved	Must be read as 0.

Bit(s)	Symbol	Name	Description
[15:0]	MID	Message ID	<p>Message ID.</p> <p>Pattern assigned by system software.</p> <p>The Multiple Message Enable field (bits [6:4] of the Message Control Register described in Section 9.2.31) defines the number of low-order message data bits that the function is allowed to modify to generate system software-allocated vectors. For example, when the Multiple Message Enable field is “010”, the function has been allocated four vectors and is permitted to modify message data bits 1 and 0. (A function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is “000”, the function is not permitted to modify the message data.</p>

9.2.35 PCIe Capabilities Identifier Register

Mnemonic: PCIE_CAP_ID

Configuration Address: 0x70

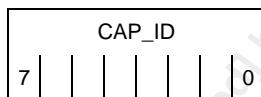
Memory Offset: 0x370

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x10

Description: Indicates the PCIe Capability structure. This field must return a Capability ID of 0x10 indicating that this is a PCIe Capability Structure.



9.2.36 PCIe Next Capability Register

Mnemonic: PCIE_NEXT_CAP

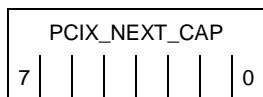
Configuration Address: 0x71

Size: 8 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0xAC

Description: Pointer to the next item in the capabilities list.



9.2.37 PCIe Capabilities Register

Mnemonic: PCIE_CAP

Configuration Address: 0x72

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x0002

Description: Identifies the PCIe device type and associated capabilities.

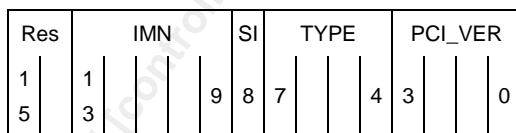


Table 206 PCIe Capabilities Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[15:14]	Res	Reserved	Reserved	0x0	RO

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[13:9]	IMN	Interrupt Message Number	This device only generates at most one message.	5'b00000	RO
8	SI	Slot Implemented	Does not apply to endpoint.	0x0	RO
[7:4]	TYPE	Device/Port Type	Indicates a PCIe endpoint device.	0x0	RO
[3:0]	PCI_VER	Capability Version	Indicates PCI-SIG defined PCIe Capability structure version number. The value is 0x01 for this specification.	0x2	RO

9.2.38 Device Capabilities Register

Mnemonic: DEVICE_CAP

Configuration Address: 0x74

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00008701

Description: Identifies the PCIe device specific capabilities.

Res	PLC	PLV	Res	R B E R	P I P 4	A I P 3	A B P 2	L1AL	L0AL	E T- F S	PhF S 3	MPSS																													
3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	2 9	2 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	1 9	8 7	7 6	6 5	5 4	4 3	3 2	2 1	1 0	0 0

Table 207 Device Capabilities Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:28]	Res	Reserved	Reserved	0x0	RO
[27:26]	PLC	Captured Slot Power Limit Scale	Not supported.	0x00	RO
[25:18]	PLV	Captured Slot Power Limit Value	Not supported.	0x00	RO
[17:16]	Res	Reserved	Reserved	0x0	RO

Bit(s)	Symbol	Name	Description	Reset State	Attributes
15	RBER	Role Based Error Reporting	Role-based error reporting for ECN.	0x1	RO
14	PIP	Power Indicator Present	Not supported.	0x0	RO
13	AIP	Attention Indicator Present	Not supported.	0x0	RO
12	ABP	Attention Button Present	Not supported.	0x0	RO
[11:9]	L1AL	Endpoint L1 Acceptable Latency	This field describes how much latency an Endpoint can withstand when it transitions from the L1 Link state to the L0 Link state (> 64 µs).	0x3	RO
[8:6]	L0AL	Endpoint L0 Acceptable Latency	This field describes how much latency an Endpoint can withstand when it transitions from the L0s Link state to the L0 Link state (2 µs to 4 µs).	0x4	RO
5	ETFS	Extended Tag Field Supported	Indicates the maximum supported size of the Tag field as a requester. 0: 5-bit Tag field supported (the only size supported by the SPC 8x6G).	0x0	RO
[4:3]	PhFS	Phantom Functions Supported	Phantom functions not supported.	0x0	RO
[2:0]	MPSS	Maximum Payload Size Supported	Indicates the maximum payload size that the device can support for TLPs. The chip supports a maximum payload size of 256 bytes.	0x1	RO

9.2.39 Device Control Register

Mnemonic: DEVICE_CTRL

Configuration Address: 0x78

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x2810

Description: Controls PCIe device specific parameters.

See “Error Information” section.

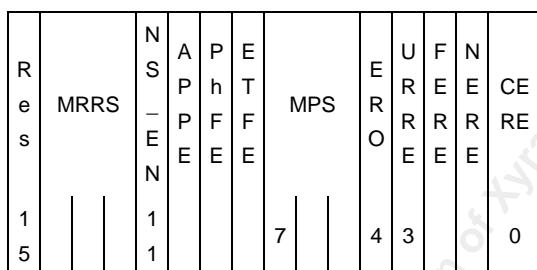


Table 208 Device Control Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
15	Res	Reserved	Reserved		
[14:12]	MRRS	Maximum Read Request Size	Sets the maximum read request size for the device as a requester. Defined encodings for this field are: 000: 128 bytes maximum read request size 001: 256 bytes maximum read request size 010: 512 bytes maximum read request size 011: 1024 bytes maximum read request size 100: 2048 bytes maximum read request size 101: 4096 bytes maximum read request size 110: Reserved 111: Reserved	0x2	RW
11	NS_EN	Enable No Snoop	When set to 1, the device is permitted to set the No Snoop bit in the requester's attributes of transactions it initiates that do not require hardware enforced cache coherency.	0x1	RW
10	APPE	Auxiliary Power PM Enable	Not supported.	0x0	RO

Bit(s)	Symbol	Name	Description	Reset State	Attributes
9	PhFE	Phantom Functions Enable	The device does not support Phantom Functions.	0x0	RO
8	ETFE	Extended Tag Field Enable	The device is restricted to a 5-bit tag field	0x0	RO
[7:5]	MPS	Maximum Payload Size	<p>This field sets maximum TLP payload size for the device. Permissible values that can be programmed are indicated by the Maximum Payload Size Supported field in the Device Capabilities Register.</p> <p>Valid encodings for this field are:</p> <ul style="list-style-type: none"> 000: 128 bytes maximum payload size 001: 256 bytes maximum payload size 010: 512 bytes maximum payload size 011: 1024 bytes maximum payload size 100: 2048 bytes maximum payload size 101: 4096 bytes maximum payload size <p>The SPC 8x6G supports up to 4096 bytes maximum payload in PCIe TLPs.</p> <p>NOTE: The MPS field is only reset at power-on or data link down reset.</p> <p>NOTE: Only writes to Function 0 affect the behavior of the SPC 8x6G.</p>	0x0	RW
4	ERO	Enable Relaxed Ordering	Relaxed ordering is supported	0x1	RO
3	URRE	Unsupported Request Reporting Enable	<p>When set, this bit enables the reporting of Unsupported Requests to the root complex by means of an ERR_NONFATAL message.</p> <p>Non-posted requests that are unsupported are always completed with a completion status of UR, as required by the PCIe Specification 1.0a.</p>	0x0	RW
2	FERE	Fatal Error Reporting Enable	<p>When set, this bit enables reporting of fatal errors to the root complex. Also see SERR_EN in the PCIe Configuration Command Register.</p>	0x0	RW
1	NERE	Non-Fatal Error Reporting Enable	<p>When set, this bit enables reporting of non-fatal errors to the root complex. Also see SERR_EN in the PCIe Configuration Command Register.</p>	0x0	RW
0	CERE	Correctable Error Reporting Enable	<p>When set, this bit enables reporting of correctable errors to the root complex. Also see SERR_EN in the PCIe Configuration Command Register.</p>	0x0	RW

9.2.40 Device Status Register

Mnemonic: DEVICE STAT

Configuration Address: 0x7A

Size: 16 Bits

Access Privilege: PCI Configuration (Read-Write), PCI Memory (Read-Write)

Reset State: 0x0000

Description: Provides information about PCIe device specific parameters. See “Error Information” section.

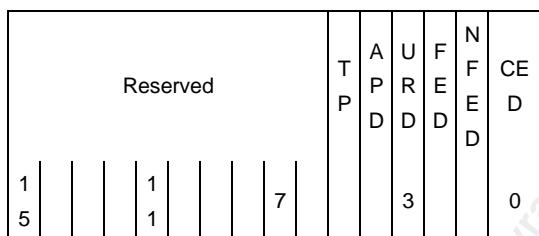


Table 209 Device Status Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[16:5]	Res	Reserved	Reserved	0x0	RO
5	TP	Transactions Pending	Set to indicate that the function has issued non-posted requests, which have not been completed.	0x0	RO
4	APD	Auxiliary Power Detected	Not supported.	0x0	RO
3	URD	Unsupported Request Detected	Indicates that the device received an unsupported request on PCIe. Errors are logged in this register regardless of whether error reporting is enabled in the Device Control Register. An unsupported request also sets the NFED bit.	0x0	RW1C (Read, write logic 1 to clear)
2	FED	Fatal Error Detected	Indicates that the device detected a fatal error on PCIe. Errors are logged in this register regardless of whether error reporting is enabled in the Device Control Register.	0x0	RW1C

Bit(s)	Symbol	Name	Description	Reset State	Attributes
1	NFED	Non-Fatal Error Detected	Indicates a PCIe non-fatal error has been detected. Errors are logged in this register regardless of whether error reporting is enabled in the Device Control Register.	0x0	RW1C
0	CED	Correctable Error Detected	Indicates a PCIe correctable error was detected. Errors are logged in this register regardless of whether error reporting is enabled in the Device Control Register.	0x0	RW1C

9.2.41 Link Capabilities Register

Mnemonic: LINK_CAP

Configuration Address: 0x7C

Size: 32 Bits

Access Privilege: PCI Configuration (Read-Only), PCI Memory (Read-Only)

Reset State: 0x00033C82

Description: Identifies the PCIe link specific capabilities.

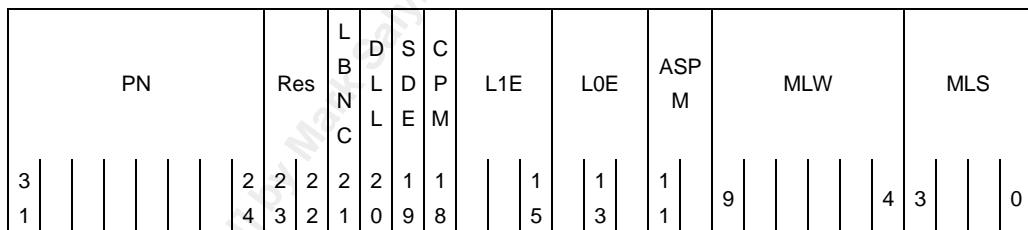


Table 210 Link Capabilities Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:24]	PN	Port Number	Indicates the PCIe Port number for the given PCIe Link.	0x00	RO
[23:22]	Res	Reserved	Reserved	0x0	RO
[21]	LBNC	LNK_BW_NOTIFY_CAP	Link Bandwidth Notification Capability	0x0	RO
[20]	DLL	DATA_LN_K_LYR	Data Link Layer Active Reporting Capable,	0x0	RO

Bit(s)	Symbol	Name	Description	Reset State	Attributes
19	SDE	SURPRISE_DWN_ERR	Surprise Down Error Reporting Capable not supported	0x0	RO
18	CPC	CLK_PM_CAP	Clock Power Management	0x0	RO
[17:15]	L1E	L1 Exit Latency	L1 Exit Latency	0x6	RO
[14:12]	L0E	L0 Exit Latency	L0 Exit Latency	0x3	RO
[11:10]	ASPM	Active State Link PM Support	Describes the level of active power management supported on the given PCIe Link.	0x3	RO
[9:4]	MLW	Maximum Link Width	Signal from PCIe core.	0x08	RO
[3:0]	MLS	Maximum Link Speed	The only defined encoding is: 0001b: 2.5 Gbit/s link 0010b: 5.0 Gbit/s link	0x2	RO

9.2.42 Link Control Register

Mnemonic: **LINK_CTRL**

Configuration Address: 0x80

Size: 16 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x0000

Description: Controls PCIe link specific parameters.

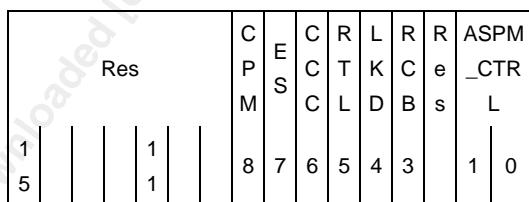


Table 211 Link Control Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[15:9]	Res	Reserved	Reserved	0x00	RO
8	CPM	Clock Power Management Enable	Not supported. Read as 0.	0x0	RO
7	ES	Extended Synch	0: Normal synchronization sequence. 1: Extended synchronization sequence.	0x0	RW
6	CCC	Common Clock Configuration	Not implemented.	0x0	RW
5	RL	RETRAIN_LINK	Not applicable for Endpoint device.	0x0	RO
4	LD	LINK_DISABLE	Not applicable for Endpoint device.	0x0	RO
3	RCB	Read Completion Boundary	May be set by configuration software to indicate the RCB value of the Root Port upstream from the endpoint. Defined encodings are: 0b: 64 bytes 1b: 128 bytes	0x0	RW
2	Res	Reserved	Reserved	0x0	RO
[1:0]	ASPM_CTRL	Active State Link PM Control	Describes the level of active power management supported on the given PCIe Link.	0x00	RW

9.2.43 Link Status Register

Mnemonic: **LINK_STAT**

Configuration Address: 0x82

Size: 16 Bits

Access Privilege: PCI Configuration (Read Only), PCI Memory (Read Only)

Reset State: Varies

Description: Provides PCIe link specific parameters.

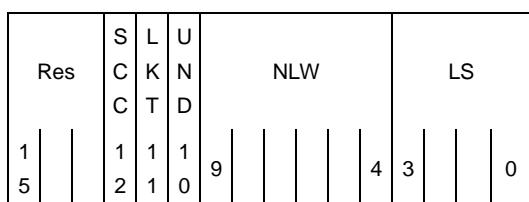


Table 212 Link Status Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[15:13]	Res	Reserved	Reserved	0x00	RO
12	SCC	Slot Clock Configuration	0: The function uses an independent reference clock rather than a reference clock on the connector. 1: The function uses the same physical reference clock that the platform provides on the connector.	Function-specific	RO
11	LKT	Link Training	Not applicable for an upstream Port or Endpoint device. Read as 0.	0x0	RO
10	UND	Undefined	Undefined for PCI Express 1.1/2.0.	0x0	RO
[9:4]	NLW	Negotiated Link Width	This field describes the negotiated width of the given PCIe Link. Defined encodings are: 000001b x1 000100b x4 001000b x8	Function-specific	RO
[3:0]	LS	Link Speed	This field describes the link speed of the given PCIe Link. The only defined encoding is: 0001b: 2.5 Gbit/s PCIe Link 00010b: 5.0 Gbit/s PCIe Link	0x1	RO

9.2.44 MSI-X Capabilities Register

Mnemonic: MSIX CAP

Configuration Address: 0xAC

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x000F0011

Description: PCIe MSI-X capability structure register.

MSIX _EN	FN_MS_K	Res	TBL_SZ	NXT_PTR	CAP_ID
31	30	2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0

Table 213 MSI-X Capabilities Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
31	MSIX_EN	MSI-X Enable	If 1 and if the MSI Enable bit in the MSI Message Control Register is 0, the function is permitted to use MSI-X to request service and is prohibited from using its INTx# pin.	0x0	RW
30	FN_MSK	Function Mask	If 1, all the vectors associated with the function are masked, regardless of what their per-vector mask bits state. If 0, each vector's mask bit determines whether the vector is masked or not.	0x0	RW
[29:27]	Res	Reserved	Reserved	0x0	RO
[26:16]	TBL_SZ	Table Size	Indicates the number of vectors. A value of 15 means there are 16 vectors.	0x00F	RO
[15:8]	NXT_PTR	Next Pointer	Pointer to the next item in the capabilities list. A value of 0 indicates there are no more items.	0x00	RO
[7:0]	CAP_ID	Capability ID	The value of 0x11 in this field identifies the function as being MSI-X capable.	0x11	RO

9.2.45 Table Offset Register

Mnemonic: **TBL_OFFSET**

Configuration Address: **0xB0**

Size: **32 Bits**

Access Privilege: **PCI Configuration (Read Only), PCI Memory (Read Only)**

Reset State: **0x00002000**

Description: **Table offset and BIR (BAR Indicator Register) for MSI-X Capability.**

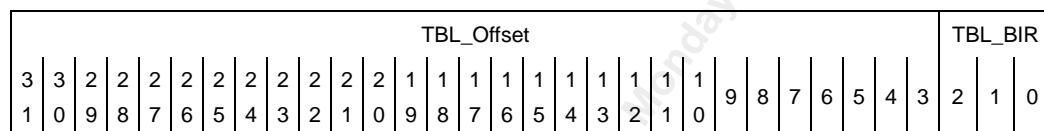


Table 214 Offset Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:3]	TBL_Offset	Table Offset	Used as an offset from the base address.	0x400	RO
[2:0]	TBL_BIR	Table BIR	Indicates which function's Base Address register set is used to map the function's MSI-X Table into memory space. MEMBase Lower is at 0x020. The BARs start from 0x10 in the configuration space. BIR Value Base Address 0 0x10 1 0x14 2 0x18 3 0x1C 4 0x20 5 0x24 6 Reserved 7 Reserved	0x0	RO

9.2.46 PBA Offset Register

Mnemonic: PBA_OFFSET

Configuration Address: 0xB4

Size: 32 Bits

Access Privilege: PCI Configuration (Read Only), PCI Memory (Read Only)

Reset State: 0x00004000

Description: PBA offset and BIR (BAR Indicator Register) for MSI-X Capability.

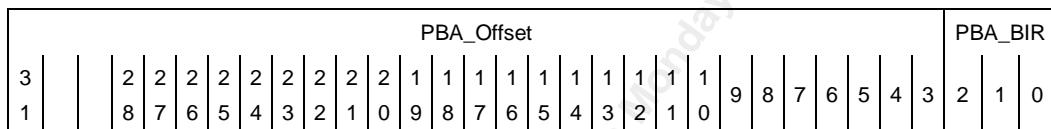


Table 215 PBA Offset Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:3]	PBA_Offset	PBA Offset	Used as an offset from the base address.	0x800	RO
[2:0]	PBA_BIR	PBA BIR	Indicates which Base Address register function is used to map the function's MSI-X PBA into memory space. The BARs start from 0x10 in the configuration space. BIR Value Base Address 0 0x10 1 0x14 2 0x18 3 0x1C 4 0x20 5 0x24 6 Reserved 7 Reserved	0x0	RO

9.2.47 Enhanced Capability Header Register

Mnemonic: PCIE_CAP_HD

Configuration Address: 0x100

Size: 32 Bits

Access Privilege: PCI Configuration (Read Only), PCI Memory (Read Only)

Reset State: 0x00010001

Description: Enhanced capability header.

NXT_CAP_Offset										CAP_VER				EXT_CAP_ID																
3		2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	
1		8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

Table 216 Enhanced Capability Header Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:20]	NXT_CAP_Offset	Next Capability Offset	Next Capability Offset	0x000	RO
[19:16]	CAP_VER	Capability Version	Capability Version	0x1	RO
[15:0]	EXT_CAP_ID	Extended Capability ID	This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability.	0x0001	RO

9.2.48 Uncorrectable Error Status Register

Mnemonic: UE STAT

Configuration Address: 0x104

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x00000000

Description: This register reports error status of individual error sources on a PCIe device. Software may clear an error status by writing a 1 to the respective bit. Writing a 0 has no effect.

Table 217 Uncorrectable Error Status Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:21]	Res	Reserved	Reserved	0x000	R0
20	URS	Unsupported Request Error Status	This bit reports the unsupported request error status.	0x0	RW1CS ¹
19	ECRC_ERR_STS	ECRC Error Status	This bit reports the ECRC error status.	0x0	RW1CS
18	MTS	Malformed TLP Status	This bit reports the malformed TLP status.	0x0	RW1CS ¹
17	RX_OVERFL_W_STS	Receiver Overflow Status	This bit reports the receiver overflow status.	0x0	RW1CS
16	UCS	Unexpected Completion Status	This bit reports the unexpected completion status.	0x0	RW1CS ¹
15	CAS	Completer Abort Status	This bit reports the completer abort status.	0x0	RW1CS ¹

Bit(s)	Symbol	Name	Description	Reset State	Attributes
14	CTS	Completion Timeout Status	This bit reports the completion timeout status. Note: Completion timeout value is 8.5-10 ms.	0x0	RW1CS ¹
13	FPS	Flow Control Protocol Error Status	This bit reports the flow control protocol error status.	0x0	RO
12	PTS	Poisoned TLP Status	This bit reports the poisoned TLP status.	0x0	RW1CS ¹
[11:6]	Res	Reserved	Reserved	0x00	RO
5	SDS	Surprise Down Error Status	Not supported	0x0	RO
4	DLS	Data Link Protocol Error Status	This bit reports the data link protocol error status.	0x0	RW1CS ¹
[3:1]	Res	Reserved	Reserved	0x0	RO
0	UND	Undefined	This bit is undefined for PCI Express 1.1/2.0.	0x0	RO

Note

- RW1CS is a designation for: Read, Write “1” to Clear, Sticky (not initialized or modified by reset).

9.2.49 Uncorrectable Error Mask Register

Mnemonic: UE_MASK

Configuration Address: 0x108

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x00000000

Description: This register controls reporting of individual errors by the device to the root complex.

Res																Res				U
R	C	T	O	C	A	T	P	T						S	D	L		U		
M	M	M	M	M	M	M	M	M						D	M	M		N		
3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	0	
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1
																				0

Table 218 Uncorrectable Error Mask Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:21]	Res	Reserved	Reserved	0x000	RO
20	URM	Unsupported Request Error Mask	This bit controls the unsupported request error mask.	0x0	RWS
19	ECM	ECRC Error Mask	This bit controls the ECRC error mask.	0x0	RWS
18	MTM	Malformed TLP Error Mask	This bit controls the malformed TLP error mask.	0x0	RWS
17	ROM	Receiver Overflow Error Mask	This bit controls the receiver overflow error mask.	0x0	RWS
16	UCM	Unexpected Completion Error Mask	This bit controls the unexpected completion error mask.	0x0	RWS
15	CAM	Completer Abort Error Mask	This bit controls the completer abort error mask.	0x0	RWS
14	CTM	Completion Timeout Error Mask	This bit controls the completion timeout error mask.	0x0	RWS

Bit(s)	Symbol	Name	Description	Reset State	Attributes
13	FPM	Flow Control Protocol Error Mask	This bit controls the flow control protocol error mask.	0x0	RWS
12	PTM	Poisoned TLP Mask	This bit controls the poisoned TLP mask.	0x0	RWS
[11:6]	Res	Reserved	Reserved	0x0	RO
5	SDM	Surprise Down Error Mask	Not supported	0x00	RO
4	DLM	Data Link Protocol Error Mask	This bit controls the data link protocol error mask.	0x0	RWS
[3:1]	Res	Reserved	Reserved	0x0	RO
0	UND	Undefined	This bit is undefined for PCI Express 1.1/2.0.	0x0	RO

9.2.50 Uncorrectable Error Severity Register

Mnemonic: UE_SEV

Configuration Address: 0x10C

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x00062030

Description: This register controls whether an individual error is reported as a non-fatal (when the corresponding bit is clear) or a fatal error (when the corresponding bit is set). Bits are not initialized or modified by reset.

Res										U R V	E C V	M T V	R O V	U C V	C A V	C T V	F P V	P T V	Res				S D S D	D L V V	Res		U N D 0				
3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9 9	8 8	7 7	6 6	5 5	4 4	3 3	2 2	1 1	0 0

Table 219 Uncorrectable Error Severity Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:21]	Res	Reserved	Reserved	0x000	R0
20	URV	Unsupported Request Error Severity	This bit controls the unsupported request error severity.	0x0	RWS
19	ECV	ECRC Error Severity	This bit controls the ECRC error severity.	0x0	RWS
18	MTV	Malformed TLP Severity	This bit controls the malformed TLP severity.	0x1	RWS
17	ROV	Receiver Overflow Severity	This bit controls the receiver overflow severity.	0x1	RWS
16	UCV	Unexpected Completion Severity	This bit controls the unexpected completion severity.	0x0	RWS
15	CAV	Completer Abort Severity	This bit controls the completer abort severity.	0x0	RWS
14	CTV	Completion Timeout Severity	This bit controls the completion timeout severity.	0x0	RWS
13	FPV	Flow Control Protocol Error Severity	This bit controls the flow control protocol error severity.	0x1	RWS
12	PTV	Poisoned TLP Severity	This bit controls the poisoned TLP severity.	0x0	RWS
[11:6]	Res	Reserved	Reserved	0x00	RO
5	SDV	Surprise Down Error Severity	Not supported	0x1	RO
4	DLV	Data Link Protocol Error Severity	This bit controls the data link protocol error severity.	0x1	RWS
[3:1]	Res	Reserved	Reserved	0x0	RO
0	UND	Undefined	This bit is undefined in PCI Express 1.1/2.0.	0x0	RO

9.2.51 Correctable Error Status Register

Mnemonic: CE_STAT

Configuration Address: 0x110

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x00000000

Description: This register reports the status of individual correctable error sources on a PCIe device. Software may clear an error status by writing a 1 to the respective bit. Writing a 0 has no effect.

Res																AN	R	Res			R	B	B	Res			RX				
3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	13	1	1	1	9	8	7	6	5	4	3	2	1	0		
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	13	2	1	0	9	8	7	6	5	4	3	2	1	0

Table 220 Correctable Error Status Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:14]	Res	Reserved	Reserved	0x0000	RO
13	ANFES	Advisory Non-Fatal Error Status		0x0	RW1CS ¹
12	RTS	Replay Timer Timeout Status		0x0	RW1CS ¹
[11:9]	Res	Reserved	Reserved	0x0	RO
8	RNS	Replay Number Rollover Status		0x0	RW1CS ¹
7	BDS	Bad DLLP Status		0x0	RW1CS ¹
6	BTS	Bad TLP Status		0x0	RW1CS ¹
[5:1]	Res	Reserved	Reserved	0x00	RO
0	RXS	Receiver Error Status		0x0	RW1CS ¹

Note

- o RW1CS is a designation for: Read, Write “1” to Clear, Sticky (not initialized or modified by reset).

9.2.52 Correctable Error Mask Register

Mnemonic: CE_MASK

Configuration Address: 0x114

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x00002000

Description: This register controls reporting of individual correctable errors by the device to the root complex. Bits are not initialized or modified by reset.

Res																AN	R	Res			R	B	B	Res			R				
3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	AN	R	Res			R	N	D	B	T	Res			R
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	FE	T	Res			N	M	M	M	T	Res			X
																		13	12	1	1	9	8	7	6	5	4	3	2	1	0

Table 221 Correctable Error Mask Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:14]	Res	Reserved	Reserved	0x0000	R0
13	ANFEM	Advisory Non-Fatal Error Mask	This bit controls the advisory non-fatal error mask.	0x1	RWS ⁶
12	RTM	Replay Timer Timeout Mask	This bit controls the replay timer timeout mask.	0x0	RWS
[11:9]	Res	Reserved	Reserved	0x0	RO
8	RNM	Replay Number Rollover Mask	This bit controls the replay number rollover mask.	0x0	RWS
7	BDM	Bad DLLP Mask	This bit controls the bad DLLP mask.	0x0	RWS
6	BTM	Bad TLP Mask	This bit controls the bad TLP mask.	0x0	RWS
[5:1]	Res	Reserved	Reserved	0x00	RO
0	RXM	Receiver Error Mask	This bit controls the receiver error mask.	0x0	RWS

⁶RWS is a designation for: Read, Write, Sticky (not initialized or modified by reset).

9.2.53 Advanced Error Capabilities and Control Register

Mnemonic: ADV_ERR_CTRL

Configuration Address: 0x118

Size: 32 Bits

Access Privilege: PCI Configuration (Read/Write), PCI Memory (Read/Write)

Reset State: 0x000000A0

Description: This register has the control and enable for AER. It is for uncorrectable errors only. Bits are not initialized or modified by reset.

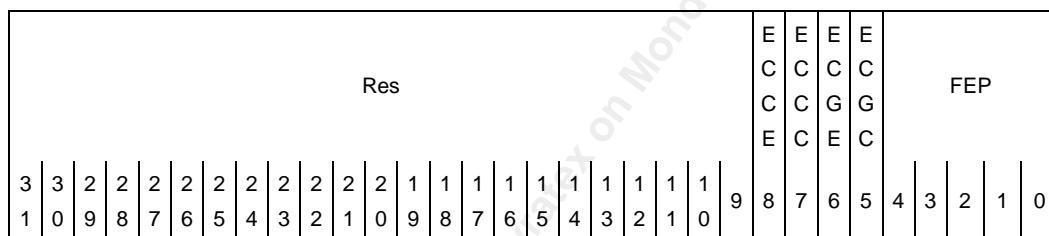


Table 222 Advanced Error Capabilities and Control Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[31:9]	Res	Reserved	Reserved	0x0000000	RO
8	ECCE	ECRC Check Enable	This bit enables ECRC checks.	0x0	RWS
7	ECCC	ECRC Check Capable	This bit reports whether ECRC checks are capable.	0x1	RO
6	ECGE	ECRC Generation Enable	This bit enables ECRC generation.	0x0	RWS
5	ECGC	ECRC Generation Capability	This bit reports whether ECRC generation is capable.	0x1	RO
[4:0]	FEP	First Error Pointer	Identifies the bit position of the first error reported in the Uncorrectable Error Status Register.	0x00	ROS

9.2.54 Header Log Register

Mnemonic: HD_LOG (DW0, DW1, DW2, DW3)

Configuration Address: 0x11C, 0x120, 0x124, 0x128

Size: 128 Bits

Access Privilege: PCI Configuration (Read Only), PCI Memory (Read Only)

Reset State: 0x0000000

Description: This register captures the header of the TLP corresponding to a detected uncorrectable error. Bits are not initialized or modified by reset.

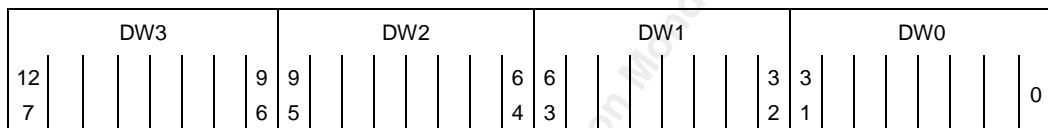


Table 223 Header Log Register Bits

Bit(s)	Symbol	Name	Description	Reset State	Attributes
[127:96]	DW3	DW3 (byte12, byte13, byte14, byte15)		0x0	ROS
[95:64]	DW2	DW2 (byte8, byte9, byte10, byte11)		0x0	ROS
[63:32]	DW1	DW1 (byte4, byte5, byte6, byte7)		0x0	ROS
[31:0]	DW0	DW0 (byte0, byte1, byte2, byte3)		0x0	ROS

10 Chip Registers

10.1.1 Configuration Modes

The SPC 8x6G chip registers are mapped into two 64-bit and one 32-bit wide memory BAR regions. For details about programming using the SPC 8x6G memory space see Section 2.6, “Memory Address Space” and the MEMBASE registers in Sections 9.2.14, 9.2.15, 9.2.16, and 9.2.17.

Table 224 Chip Register Configuration Modes

BAR	MEMBASE	Size	Width	Use	Document References
0x10 (0)	MEMBASE-I	64 KB	64 bits	Contains the registers used primarily for message passing.	Message Passing: Section 10.2
0x18 (2)	MEMBASE-II	64 KB	64 bits	Contains the PCS/PMA chip registers and some of the top-level registers.	Top-level: Section 10.3 PCS/PMA: Section 10.4
0x20 (4)	MEMBASE-III	64 KB	32 bits	Contains the rest of the chip registers.	Top-level: Section 10.3 OSSP: Section 10.5 SAS/SATA PHY Layer: Section 10.6 SAS/SATA Port Adapter Link: Section 10.7 GSM: Section 10.8 MBIC: Section 10.9 Miscellaneous Registers: Section 10.10

10.2 Messaging Unit Registers

The 64-bit BAR specified at PCI configuration address 0x10 (MEMBASE-I) contains the following doorbell and scratchpad registers. The following subsections provide the descriptions for these registers. See Section 5.1, “MPI Configuration Table Access” and Section 5.2.5, “MPI SAS PHY Analog Setup Table Fields” for further details of how these registers are used when reading and writing to the configuration table during initialization.

Table 225 Messaging Unit Address Map

Offset (Hex)	Name	Access
0x04	Inbound Doorbell Register	Host R/W Local INT

Offset (Hex)	Name	Access
0x20	Inbound Doorbell Clear Register	Host No access Local R/W
0x3C	Outbound Doorbell Register	Host RO Local R/W
0x40	Outbound Doorbell Clear Register	Host R/W Local INT
0x44	Scratchpad 0 Register	Host RO Local R/W
0x48	Scratchpad 1 Register	Host RO Local R/W
0x4C	Scratchpad 2 Register	Host RO Local R/W
0x50	Scratchpad 3 Register	Host RO Local R/W
0x54	Host Scratchpad 0 Register	Host R/W Local RO
0x58	Scratchpad 1 Register	Host R/W Local RO
0x5C	Host Scratchpad 2 Register	Host R/W Local RO
0x60	Host Scratchpad 3 Register	Host R/W Local RO
0x64	Host Scratchpad 4 Register	Host R/W Local RO
0x68	Host Scratchpad 5 Register	Host R/W Local RO
0x6C	Host Scratchpad 6 Register	Host R/W Local RO
0x70	Host Scratchpad 7 Register	Host R/W Local RO
0x74	Outbound Doorbell Mask Register	Host R/W Local R/W

10.2.1 Inbound Doorbell Register

Mnemonic: Inbound Doorbell Register

Configuration Address: 0x04

Size: 32 Bits

Access Privilege: Host R/W, Local INT

Reset State: 0x0

Description: This register is set by the host when initiating a messaging interface configuration-related operation.

This register is also indirectly set by the SPC 8x6G and read by the host during a messaging interface configuration-related operation as part of the host-to-SPC 8x6G synchronization. During this operation, each of the register bits is toggled by the SPC 8x6G as a means for host-to-SPC 8x6G synchronization.

Table 226 Inbound Doorbell Register

Bit(s)	Access	Name	Description	Reset State
[31:4]	See above	Res	Reserved	See above
3	See above	MPIIU	When the host sets this bit, it is used for initiating Host-SPC 8x6G MPI Inbound Un-freeze (Section 5.2.6.4). The host reads and polls for this bit to be cleared during synchronization in Host-SPC 8x6G MPI Inbound Un-freeze .	See above
2	See above	MPIIF	When the host sets this bit, it is used for initiating Host-SPC 8x6G MPI Inbound Freeze (Section 5.2.6.3). The host reads and polls for this bit to be cleared during synchronization in Host-SPC 8x6G MPI Inbound Freeze .	See above
1	See above	MPICT	When the host sets this bit, it is used for initiating Host-SPC 8x6G MPI Communication Termination (Section 5.2.6.2). Host reads and polls for this bit to be cleared during synchronization in Host-SPC 8x6G MPI Communication Termination .	See above
0	See above	MPIINI	When the host sets this bit, it is used for initiating Host-SPC 8x6G MPI Initialization (Section 5.2.6.1). Host reads and polls for this bit to be cleared during synchronization in Host-SPC 8x6G MPI Initialization .	See above

10.2.2 Inbound Doorbell Clear Register

Mnemonic: Inbound Doorbell Clear Register

Configuration Address: 0x20

Size: 32 Bits

Access Privilege: Host No access, Local R/W

Reset State: 0x0

Description: This register is only to be used by SPC 8x6G firmware. The host should not use this register. The host reads will always read 0x00000000 from this register.

This register is set by the SPC 8x6G to clear the bit(s) in Inbound Doorbell Register.

Table 227 Inbound Doorbell Clear Register

Bit(s)	Access	Description	Reset State
[31:4]	See above	Reserved	See above
3	See above	Used for synchronization in Host-SPC 8x6G MPI Inbound Unfreeze (Section 5.2.6.4)	See above
2	See above	Used for synchronization in Host-SPC 8x6G MPI Inbound Freeze (Section 5.2.6.3)	See above
1	See above	Used for synchronization in Host-SPC 8x6G MPI Communication Termination (Section 5.2.6.2).	See above
0	See above	Used for synchronization in Host-SPC 8x6G MPI Initialization (Section 5.2.6.1).	See above

10.2.3 Outbound Doorbell Register

Mnemonic:	Outbound Doorbell (ODR)
Configuration Address:	0x3C
Size:	32 Bits
Access Privilege:	Host RO, Local R/W
Reset State:	0x0
Description:	Each bit of the ODR is mapped (hard-wired) to the MSI-X (or MSI) vector table index entry. There are 16 MSI-X entries and 32 MSI vectors supported in SPC 8x6G. The upper 16 bits of the ODR will be folded into the low 16 bits for MSI-X. For legacy INT-X, any bit(s) will generate a host interrupt.

The SPC 8x6G MSGU sets the appropriate bit based on the host setting in MPI Outbound Queue Configuration Table's, *Outbound Queue n Interrupt Vector (OQIVn)* field (see Section 5.2.4, “[MPI Outbound Queue Configuration Table Fields](#)”). The *OQIVn* is a zero-based relative interrupt vector assigned to this instance of the SPC 8x6G.. A logic 0 interrupt vector is mapped to bit 0 of the ODR, a logical 1 interrupt vector is mapped to bit 1 of the ODR, etc. Through this mapping method, the host can assign a particular OQ to a specific interrupt vector.

10.2.4 Outbound Doorbell Clear Register

Mnemonic:	Outbound Doorbell Clear (ODCR)
Configuration Address:	0x40
Size:	32 Bits
Access Privilege:	Host R/W, Local R/W
Reset State:	0x0
Description:	The host sets the bit(s) in the ODCR to clear the corresponding bit(s) in the Outbound Doorbell register (ODR). For legacy INTx, it also clears (de-asserts) the host interrupt.

10.2.5 Scratchpad 0 Register

Mnemonic: Scratchpad 0

Configuration Address: 0x44

Size: 32 Bits

Access Privilege: Host RO, Local R/W

Reset State: 0x0

Description: When [Scratchpad 1 Register](#) bits [1:0], AAP_STATE, are set to 11 (ready state), this register describes the [MPI Configuration Table Access](#) (see Section [5.1](#)):

- The upper 6 bits [31:26] of the Scratchpad 0 Register describe the offset within the PCI Configuration Space where the BAR is located.
- The lower 26 bits [25:0] of the Scratchpad 0 Register describe the offset within the BAR.

When [Scratchpad 1 Register](#) bits [1:0], AAP_STATE, are set to 10 (error state), this register reports details about fatal errors in the AAP1/MSGU. The detailed fatal error code depends on the source of fatal error reported in [Scratchpad 1 Register](#) bits [31: 8], AAP_ERR. See Section [11.2](#), “[Device Specific Fatal Errors](#)” for more information.

When [Scratchpad 1 Register](#) bits [1:0], AAP_STATE, are set to 00b (power-on reset state), this register is used to describe the HDA-ILA firmware and the host driver protocol in HDA mode. During the HDA-ILA initialization process, this register is set by the SPC 8x6G HDA-ILA and read by the host:

- The upper 8 bits [31:24] of the Scratchpad 0 Register describe the HDA state or response of the HDA-ILA firmware that the host driver reads. The definition of this field is defined in Section [3.21.3](#), “[HDA ILA Protocol – Command and Response/Status](#)”.
- The lower 24 bits [23:0] of the Scratchpad 0 Register describe the offset in GSM where the host is to write the HDA firmware image.

The bit definitions are described in Section [3.21.3](#), “[HDA ILA Protocol – Command and Response/Status](#)”.

10.2.6 Scratchpad 1 Register

Mnemonic: Scratchpad 1

Configuration Address: 0x48

Size: 32 Bits

Access Privilege: Host RO, Local R/W

Reset State: 0x0

Description: This register is used to report error and status indications for the AAP1/MSGU.

Table 228 MSGU Scratchpad 1 Register

Bit(s)	Access	Name	Description	Reset State
[31: 8]	See above	AAP_ERR	This field reports fatal error indicators detected on AAP1/MSGU. It is valid only if bits [1:0] are set as 10b (error state). See Section 11.2, “Device Specific Fatal Errors” for details.	See above
[7:4]	See above	Reserved	Reserved	See above
3	See above	CPU_SOFT_RESET_RDY	Internally used by SPC 8x6G Firmware. This bit is used by AAP1 to indicate to the other internal processor that AAP1 is ready for a soft reset.	See above
2	See above	SFTRST_P_F	Toggled flag indicating soft reset progress.	See above
[1:0]	See above	AAP_STATE	Ready state for the AAP1/MSGU: 00: Power-on reset state 01: Soft reset state (This is when the firmware is currently executing the soft reset procedure as requested by the host driver.) 10: Error state 11: Ready state	See above

10.2.7 Scratchpad 2 Register

Mnemonic: Scratchpad 2

Configuration Address: 0x4C

Size: 32 Bits

Access Privilege: Host RO, Local R/W

Reset State: 0x0

Description: The Scratchpad 2 Register is used to report error and status indications for the IOP.

Table 229 MSGU Scratchpad 2 Register

Bit(s)	Access	Name	Description	Reset State
[31:10]	See above	IOP_ERR	This field reports fatal error indicators detected on the IOP. It is valid only if bits [1:0] are set as 10b (error state). See Section 11.2, "Device Specific Fatal Errors" for details.	See above
[9:4]	See above	Reserved	Reserved	See above
3	See above	CPU_SOFT_RESET_RDY	Internally used by SPC 8x6G Firmware. This bit is used by the IOP to indicate to the other internal processor that the IOP is ready for a soft reset.	See above
2	See above	HOST_SOFT_RESET_RDY	Ready indicator for the host issuing a soft reset.	See above
[1:0]	See above	IOP_STATE	Ready state for the Input/Output Processing module: 00: Power-on reset state 01: Soft reset state (This is when the firmware is currently executing the soft reset procedure as requested by the host driver.) 10: Error state 11: Ready state	See above

10.2.8 Scratchpad 3 Register

Mnemonic:	Scratchpad 3
Configuration Address:	0x50
Size:	32 Bits
Access Privilege:	Host RO, Local R/W
Reset State:	0x0
Description:	<p>When Scratchpad 2 Register bits [1:0], IOP_STATE, are set to 10b (error state), the Scratchpad 3 Register is used to indicate details about the fatal errors in the IOP.</p> <p>The detailed fatal error code depends on the source of the fatal error reported in Scratchpad 2 Register bits [31:13], IOP_ERR. See Section 11.2, “Device Specific Fatal Errors” for details.</p> <p>When Scratchpad 2 Register bits [1:0], IOP_STATE, are set to 11b (ready state) the Scratchpad 3 Register is used by firmware to indicate the state of the firmware images as well as the active firmware image as shown in Table 230.</p> <p>In HDA mode, these flags are not meaningful.</p>

Table 230 Scratchpad 3 Register when IOP_STATE is in Ready State

Bit(s)	Access	Name	Description	Reset State
[31: 4]	See above	Reserved	Reserved	See above
3	See above	IMG_FLAG_VALID	FW Image Active partition valid indicator. 0b : ACTIVE_IMG flag is not valid 1b : ACTIVE_IMG flag is valid	See above
2	See above	FW_IMG_B_VALID	FW Image B Flash partition valid indicator 0b : Image B is not valid 1b : Image B is valid	See above
1	See above	FW_IMG_A_VALID	FW Image A Flash partition valid indicator 0b : Image A is not valid	See above

Bit(s)	Access	Name	Description	Reset State
			1b : Image A is valid	
0	See above	ACTIVE_IMG	FW Image Active partition indicator. 0b : SPC 8x6G boots from Image B FW 1b : SPC 8x6G boots from Image A FW	See above

10.2.9 Host Scratchpad 0 Register

Mnemonic: Host Scratchpad 0

Configuration Address: 0x54

Size: 32 Bits

Access Privilege: Host R/W, Local RO

Reset State: 0x0

Description: This register is used as the signature for a soft reset execution. For normal mode, this soft reset signature is defined as 0x252ACBCD. For HDA mode, this soft reset signature is defined as 0xA5AA27D7.

See Section [11.4.2, “Soft Reset”](#) for details about a normal mode soft reset sequence.

See Section [11.4.3, “Soft Reset Recovery \(HDA Mode\)”](#) for details about an HDA mode soft reset sequence.

10.2.10 Host Scratchpad 1 Register

Mnemonic: Host Scratchpad 1

Configuration Address: 0x58

Size: 32 Bits

Access Privilege: Host R/W, Local RO

Reset State: 0x0

Description: This register is used for the MPI freeze and un-freeze operation for the first set of 32 IQs (IQ 0 to IQ 31). Each bit in the register represents the IQ affected by a freeze or un-freeze operation. Bit 0 represents IQ 0 and bit 31 represents IQ 31. See Section 5.2.6.3, “[Host-SPC 8x6G MPI Inbound Freeze](#)” and Section 5.2.6.4, “[Host-SPC 8x6G MPI Inbound Unfreeze](#)” for details.

10.2.11 Host Scratchpad 2 Register

Mnemonic: Host Scratchpad 2

Configuration Address: 0x5C

Size: 32 Bits

Access Privilege: Host R/W, Local RO

Reset State: 0x0

Description: This register is used for MPI freeze and un-freeze operations for the second set of 32 IQs (IQ 32 to IQ 63). Each bit in the register represents the IQ affected by a freeze or un-freeze operation. Bit 0 represents IQ 32 and bit 31 represents IQ 63. See Section 5.2.6.3, “[Host-SPC 8x6G MPI Inbound Freeze](#)” and Section 5.2.6.4, “[Host-SPC 8x6G MPI Inbound Unfreeze](#)” for details.

10.2.12 Host Scratchpad 3 Register

Mnemonic: Host Scratchpad 3

Configuration Address: 0x60

Size: 32 Bits

Access Privilege: Host R/W, Local RO

Reset State: 0x0

Description: This register is used for the host and the SPC 8x6G ILA synchronization during the HDA initialization process. The host writes to this register for the HDA-ILA command.

- The upper 8 bits [31:24] of the Host Scratchpad 3 Register describe the host command that the SPC 8x6G HDA-ILA firmware reads.
- The lower 24 bits [23:0] of the Host Scratchpad 3 Register describe the size of the image that the host writes to the GSM.

The bit definitions are described in Section 3.21.3, “[HDA ILA Protocol – Command and Response/Status](#)”.

10.2.13 Host Scratchpad 4 Register

Currently not used.

10.2.14 Host Scratchpad 5 Register

Currently not used.

10.2.15 Host Scratchpad 6 Register

Currently not used.

10.2.16 Host Scratchpad 7 Register

Currently not used.

10.2.17 Outbound Doorbell Mask Register

Mnemonic: Outbound Doorbell Mask (ODMR)

Configuration Address: 0x74

Size: 32 Bits

Access Privilege: Host R/W, Local R/W

Reset State: 0x0

Description: The host sets the bit(s) in the ODMR to mask the host interrupts that would otherwise be caused by the respective bit being set in the Outbound Doorbell register (ODR).

10.3 SPC 8x6G Top-level Registers

The 32-bit BARs specified at PCI configuration address 0x18 (MEMBASE-II) and 0x20 (MEMBASE-III) contain the following top-level SPC 8x6G registers. The following subsections provide the descriptions for these registers.

Table 231 SPC 8x6G Top-Level Address Map

Offset	MEMBASE	Name
0x0	MEMBASE-III	SPC Reset Register
0x8	MEMBASE-III	Boot Strapping Bit Register
0x20	MEMBASE-III	Device ID Register
0x24	MEMBASE-III	Device Revision Register
0x34	MEMBASE-III	SAS CSU Lock Detect Monitor Status Register
0x44	MEMBASE-III	Device Test Register
0x58	MEMBASE-III	Device LCLK Generation
0x3040	MEMBASE-II	PCIe Event Interrupt Enable Register
0x3044	MEMBASE-II	PCIe Event Interrupt Register
0x3048	MEMBASE-II	PCIe Error Interrupt Enable Register
0x304C	MEMBASE-II	PCIe Error Interrupt Register
0x3258	MEMBASE-II	Inbound AXI Translation Lower Address – Window 2 Register
0x335C	MEMBASE-II	Message Unit Outbound Doorbell Auto Clear Register
0x33C0	MEMBASE-II	Interrupt Coalescing Timer Register
0x33C4	MEMBASE-II	Interrupt Coalescing Control Register

10.3.1 Boot Strapping Bit Register

Mnemonic: TOP_BOOT_STRAP

Address: (via MEMBASE-III): 0x8

Description:

Table 232 Boot Strapping Bit Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:27]	—	UNUSED		X XXXX
[26:0]	R	STRAP_BIT	Strapping pin captured value when the device comes out of reset.	010 1100 0000 1010 0110 1000 0010

10.3.2 Device ID Register

Mnemonic: TOP_DEVICE_ID

Address: (via MEMBASE-III): 0x20

Description:

Table 233 Device ID Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	—	UNUSED		XXXX XXXX XXXX XXXX
[15:0]	R	DEVICE_ID	DEVICE_ID[15:0] provides the device identification number for the SPC 8x6G. DEVICE_ID[15:0] is set to 0x8001.	1000 0000 0000 0001

10.3.3 Device Revision Register

Mnemonic: TOP_DEVICE_REV

Address: (via MEMBASE-III): 0x24

Description:

Table 234 Device Revision Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:4]	—	UNUSED		XXXX XXXX XXXX XXXX XXXX XXXX XXXX
[3:0]	R	DEVICE_REVISION	This field contains the revision number for this device. The initial revision is 0x00. It is incremented for every subsequent device revision.	0010

10.3.4 SAS CSU Lock Detect Monitor Status Register

Mnemonic: TOP_SAS_LOCK_DET_STATUS

Address: (via MEMBASE-III): 0x34

Description:

Table 235 SAS CSU Lock Detect Monitor Interrupt Status Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:2]	—	UNUSED		XX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
1	R	LOCK_SAS_SCSU_V	The SAS SCSU Lock status bit indicates whether SAS SCSU is currently locked with the reference clock, SAS_REFCLK_C or SAS_REFCLK_P/N. When read as logic: 1: SAS SCSU is locked with SAS_REFCLK_C or SAS_REFCLK_P/N. 0: SAS SCSU is not successfully locked with SAS_REFCLK_C.	1

Bit(s)	Access	Name	Description	Reset State
0	R	LOCK_SAS_CSU_V	The SAS CSU Lock status bit indicates whether SAS CSU is currently locked with the reference clock, SAS_REFCLK_C or SAS_REFCLK_P/N. When read as logic: 1: SAS CSU is locked with SAS_REFCLK_C or SAS_REFCLK_P/N. 0: SAS CSU is not successfully locked with SAS_REFCLK_C.	1

10.3.5 Device Test Register

Mnemonic: TOP_DEVICE_TST

Address: (via MEMBASE-III): 0x44

Description:

Table 236 Device Test Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:1]	—	UNUSED		XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
0	R/W	HIZ	The HIZ bit controls the tri-state modes of the device. When the HIZ bit is logic 1, all digital output pins of the device are held in tri-state.	0

10.3.6 Device LCLK Generation

Mnemonic: DEVICE_LCLK_GENERATION

Address: (via MEMBASE-III): 0x58

Description: The host uses this register to get PHY error counts.

Table 237 Device Test Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:8]	—	UNUSED		XXXX XXXX XXXX XXXX XXXX XXXX
7	R/W	LCLK	This is the Asynchronous LCLK Request bit. Writing logic 1 to this register bit will generate a pulse on an internal LCLK signal to all the internal blocks in the device. A rising edge on the internal LCLK signal causes the values of all performance monitor counters to be latched into shadow registers so they can be read. This bit is reset automatically to logic 0 after the internal LCLK signal is asserted.	0
6	R/W	LCLK_CLEAR	When set to logic: 1: All values on the performance counters is reset to 0 after values are read. 0: All values on the performance counters is maintained after values are read.	1
[5:1]	—	UNUSED		X XXXX
0	R	TIP	When read as logic 1, a transfer to shadow register is in progress. When read as logic 0, the transfer has completed. If the corresponding SSPL_LCLK_CLEAR (per port) or ECMR_LCLK_CLEAR register bit for a specific performance monitor counter is set to 1, the counter is cleared after the transfer to the shadow registers is complete. The performance monitor counters include a Running Disparity Error counter, Invalid DWORD counter, Loss of DWORD Synchronization counter, PHY Reset Problem counter and Code Violation Error counter in the SSPL.	0

10.3.7 SPC Reset Register

Mnemonic: SPC_RESET

Address: (via MEMBASE-III): 0x00

Description: SPC 8x6G host-accessible reset control.

Table 238 SPC Reset Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	SW_DEVICE_RSTB	When logic 1, the chip is in a normal operation state. When logic 0, the entire device is going into the reset. This bit should not be used directly by customers except in the case of a fatal error.	1
[30:27]	—	UNUSED		XXXX
26	R/W	PCIE_PC_SXCBI_AR ESETN	When logic 1, the PCIe PC XCBI logic is taken out of reset. When logic 0, the PCIe PC XCBI logic will be in reset. This action sets the status and control registers to their default states.	1
25	R/W	PMIC_CORE_RSTB	When logic 1, the PMIC top-level core is taken out of reset. When logic 0, the PMIC top-level core will be in reset.	1
24	R/W	PMIC_SXCBI_ARES ETN	When logic 1, the PMIC XCBI bridge is taken out of reset. When logic 0, the PMIC XCBI bridge will be in reset.	1
23	R/W	LMS_SXCBI_ARESE TN	When logic 1, the LMS XCBI logic is taken out of reset. When logic 0, the LMS XCBI logic will be in reset. This action sets the status and control registers to their default states.	1
22	R/W	PCS_SXCBI_ARESE TN	When logic 1, the PCS/PMA XCBI logic is taken out of reset. When logic 0, the PCS/PMA XCBI logic will be in reset. This action sets the status and control registers to their default states.	1
21	R/W	PCIE_SFT_RSTB	When logic 1, the PCIe core, the PCIe analog, and the PCS/PMA core is taken out of reset. When logic 0, the PCIe core, the PCIe analog, and the PCS/PMA core will be in reset. This action will have no effect on the PCIe or the PCS/PMA status and control registers.	1

Bit(s)	Access	Name	Description	Reset State
20	R/W	PCIE_PWR_RSTB	When logic 1, the PCIe core, the PCIe analog, and the PCS/PMA core is taken out of reset. When logic 0, the PCIe core, the PCIe analog, and the PCS/PMA core will be in reset. This action will have no effect on the PCIe or the PCS/PMA status and control registers.	1
19	R/W	PCIE_AL_SXCBI_AR_ESETN	When logic 1, the PCIe AL XCBI logic is taken out of reset. When logic 0, the PCIe AL XCBI logic will be in reset. This action sets the status and control registers to their default states.	1
18	R/W	BDMA_SXCBI_ARES ETN	When logic 1, the BDMA XCBI logic is taken out of reset. When logic 0, the BDMA XCBI logic will be in reset. This action sets the status and control registers to their default states.	1
17	R/W	BDMA_CORE_RSTB	When logic 1, the BDMA core is taken out of reset. When logic 0, the BDMA core will be in reset. This action will have no effect on the BDMA status and control registers.	1
16	R/W	Reserved	Reserved.	0
15:9	—	UNUSED		XXX XXXX
8	R/W	GSM_RSTB	This is a self-cleared bit. When logic 1, the GSM subsystem is taken out of reset. When logic 0, this bit resets the GSM subsystem.	1
7	R/W	PCS_RSTB	When logic 1, the entire PCS subsystem is taken out of reset. When logic 0, the entire PCS subsystem is reset. This action sets the status and control registers to their default states.	1
6	R/W	PCS_LM_RSTB	When logic 1, the PCS_LM block in the PCS subsystem is taken out of reset. When logic 0, all logic in the PCS_LM block is reset. This action sets the status and control registers to their default states.	1
5	R/W	Reserved	Reserved.	1
4	R/W	PCS_AAP1_SS_RST B	When logic 1, the AAP1_SS block in the PCS subsystem is taken out of reset. When logic 0, all logic in the AAP1_SS block is reset. This action sets the status and control registers to their default states.	1

Bit(s)	Access	Name	Description	Reset State
3	R/W	PCS_IOP_SS_RSTB	When logic 1, the IOP_SS block in the PCS subsystem is taken out of reset. When logic 0, all logic in the IOP_SS block is reset. This action sets the status and control registers to their default states.	1
2	R/W	PCS_SPBC_RSTB	When logic 1, the SPBC block in the PCS subsystem is taken out of reset. When logic 0, all logic in the SPBC block is reset. This action sets the status and control registers to their default states.	1
1	R/W	RAAE_RSTB	When logic 1, the RAAE block is taken out of reset. When logic 0, all logic in the RAAE block is reset. This action sets the status and control registers to their default states.	1
0	R/W	OSSP_RSTB	When logic 1, the OSSP subsystem is taken out of reset. When logic 0, all logic in the OSSP subsystem is reset. This action sets the status and control registers to their default states.	1

10.3.8 PCIe Event Interrupt Enable Register

Mnemonic: EVENT_INT_ENABLE

Address: (via MEMBASE-II): 0x3040

Description:

Table 239 PCIe Event Interrupt Enable Register Bits

Bit(s)	Access	Name	Description	Reset State
31:19	-	UNUSED		X XXXX XXXX XXXX
18	R/W	RCVD_PME_TURNOFF_E4	Interrupt enable for RCVD_PME_TURNOFF interrupt. 0: Interrupt is disabled. 1: Interrupt is enabled.	1
17	R/W	RCVD_PM_TO_ACK_E4	Interrupt enable for RCVD_PM_TO_ACK interrupt. 0: Interrupt is disabled. 1: Interrupt is enabled.	0

Bit(s)	Access	Name	Description	Reset State
16	R/W	RCVD_PM_PME_E4	Interrupt enable for RCVD_PM_PME interrupt. 0: Interrupt is disabled. 1: Interrupt is enabled.	0
15:12	-	UNUSED		XXXX
11	R/W	INTD_EN	Interrupt enable for INTD legacy interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
10	R/W	INTC_EN	Interrupt enable for INTC legacy interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
9	R/W	INTB_EN	Interrupt enable for INTB legacy interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
8	R/W	INTA_EN	Interrupt enable for INTA legacy interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
7	R/W	CFG_BW_MGT_E4	Interrupt enable for Configuration Bandwidth Management based interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
6	R/W	CFG_LINK_AUTO_BW_E4	Interrupt enable for Configuration Link Auto Bandwidth based interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
5:4	-	UNUSED		XX
3	R/W	D3HOT_ENTRY_E4	Interrupt enable for d3hot entry transition based interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
2	R/W	D3HOT_D0_E4	Interrupt enable for d3hot to d0 transition based interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1
1	R/W	LINK_REQ_RST_E4	Interrupt enable for link_req_rstb based interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1

Bit(s)	Access	Name	Description	Reset State
0	R/W	TRAINING_RST_E4	Interrupt enable for training_rstb based interrupt 0: Interrupt is disabled. 1: Interrupt is enabled.	1

10.3.9 PCIe Event Interrupt Register

Mnemonic: EVENT_INT_STAT

Address: (via MEMBASE-II): 0x3044

Description: This register maintains the status of non-error event interrupts that are generated by the PCIe Core or the Application Layer.

The interrupt bits are cleared by reading (if wcimode=0) or writing a '1' (if wcimode=1).

Table 240 PCIe Event Interrupt Register Bits

Bit(s)	Access	Name	Description	Reset State
31:27	-	UNUSED		X XXXX
26	R	ECC_COR_ERROR_INT_SUMMARY	Summary bit for RAM ECC Correctable Error Interrupt Status Register.	0
25	R	ECC_UNCOR_ERROR_INT_SUMMARY	Summary bit for RAM ECC Uncorrectable Error Interrupt Status Register.	0
24	R	PCIE_ERROR_INT_SUMMARY	Summary bit for PCIe Error Interrupt Status Register.	0
23:19	-	UNUSED		X XXXX
18	R/W	RCVD_PME_TURNOFF_I4	Interrupt is asserted (if enabled) whenever a PM_PME_TURNOFF message is received. (Endpoint Mode)	0
17	R/W	RCVD_PM_TO_ACK_I4	Interrupt is asserted (if enabled) whenever a PME_TO_ACK message is received. (Root Complex Mode)	0
16	R/W	RCVD_PM_PME_I4	Interrupt is asserted (if enabled) whenever a PM_PME message is received. (Root Complex Mode)	0
15	R	MU_OB_DB_CLR_IN_T_SUMMARY	Summary bit for Message Unit Outbound Doorbell Clear Interrupt Status Register.	0
14	R	MU_IB_DB_INT_SUMMARY	Summary bit for Message Unit Inbound Doorbell Interrupt Status Register.	0

Bit(s)	Access	Name	Description	Reset State
13	R	IB_WSM_INT_SUMMARY	Summary bit for Inbound WSM Interrupt Status Register.	0
12	R	Reserved		0
11	R	INTD	Interrupt is asserted (if enabled) whenever a INTD Assert message is received.	0
10	R	INTC	Interrupt is asserted (if enabled) whenever a INTC Assert message is received.	0
9	R	INTB	Interrupt is asserted (if enabled) whenever a INTB Assert message is received.	0
8	R	INTA	Interrupt is asserted (if enabled) whenever a INTA Assert message is received.	0
7	R/W	CFG_BW_MGT_I4	Interrupt is asserted (if enabled) whenever the Configuration Bandwidth Management bit is set in the PCIe Core.	0
6	R/W	CFG_LINK_AUTO_BW_I4	Interrupt is asserted (if enabled) whenever the Configuration Link Auto Bandwidth bit is set in the PCIe Core.	0
5:4	-	UNUSED		XX
3	R/W	D3HOT_ENTRY_I4	Interrupt is asserted (if enabled) whenever the PM state machine transitions to the D3hot state.	0
2	R/W	D3HOT_D0_I4	Interrupt is asserted (if enabled) whenever the PM state machine transitions from D3hot state to the D0 state.	0
1	R/W	LINK_REQ_RST_I4	Interrupt is asserted (if enabled) whenever the link_req_rst_n signal from core transitions from 1->0. This transition is a result of the link going into a link down condition.	0
0	R/W	TRAINING_RST_I4	Interrupt is asserted (if enabled) whenever the training_rst_n from core transitions from 1->0. This transition is a result of receiving a TS1/TS2 training sequence with the Hot Reset bit set.	0

10.3.10 PCIe Error Interrupt Enable Register

Mnemonic: ERROR_INT_ENABLE

Address: (via MEMBASE-II): 0x3048

Description: This register maintains the interrupt enables for the error type interrupts that can be generated by the PCIe core and the PCIe Application Layer.

Table 241 PCIe Error Interrupt Enable Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	AL_SLV_WR_DATA_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Slave Write Data Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
30	R/W	AL_SLV_RD_DATA_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Slave Read Data Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
29	R/W	AL_SLV_WR_ADDR_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Slave Write Address Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
28	R/W	AL_SLV_RD_ADDR_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Slave Read Address Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
27	R/W	AL_MSTR_WR_DATA_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Master Write Data Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
26	R/W	AL_MSTR_RD_DATA_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Master Read Data Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
25	R/W	AL_MSTR_WR_ADDR_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Master Write Address Port. 0: Interrupt Disabled 1: Interrupt Enabled	1

Bit(s)	Access	Name	Description	Reset State
24	R/W	AL_MSTR_RD_ADD_R_PAR_ERR_E6	Controls reporting (interrupt generation) of detected parity errors on the PCIE Application Layer Master Read Data Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
23	R/W	AL_SLV_RD_RSP_SLV_ERR_E6	Controls reporting (interrupt generation) of SLV_ERR response detection on the PCIE Application Layer Slave Read Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
22	R/W	AL_SLV_RD_RSP_DEC_ERR_E6	Controls reporting (interrupt generation) of DEC_ERR response detection on the PCIE Application Layer Slave Read Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
21	R/W	AL_SLV_WR_RSP_SLV_ERR_E6	Controls reporting (interrupt generation) of SLV_ERR response detection on the PCIE Application Layer Slave Write Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
20	R/W	AL_SLV_WR_RSP_DEC_ERR_E6	Controls reporting (interrupt generation) of DEC_ERR response detection on the PCIE Application Layer Slave Write Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
19	R/W	AL_MSTR_RD_RSP_SLV_ERR_E6	Controls reporting (interrupt generation) of SLV_ERR response detection on the PCIE Application Layer Master Read Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
18	R/W	AL_MSTR_RD_RSP_DEC_ERR_E6	Controls reporting (interrupt generation) of DEC_ERR response detection on the PCIE Application Layer Master Read Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
17	R/W	AL_MSTR_WR_RSP_SLV_ERR_E6	Controls reporting (interrupt generation) of SLV_ERR response detection on the PCIE Application Layer Master Write Port. 0: Interrupt Disabled 1: Interrupt Enabled	1

Bit(s)	Access	Name	Description	Reset State
16	R/W	AL_MSTR_WR_RSP _DEC_ERR_E6	Controls reporting (interrupt generation) of DEC_ERR response detection on the PCIE Application Layer Master Write Port. 0: Interrupt Disabled 1: Interrupt Enabled	1
15	R/W	PCIE_GM_CMPSR_L U_ERR_E6	Controls reporting (interrupt generation) of PCIe-AXI Bridge Master Composer Lookup Error. 0: Interrupt Disabled 1: Interrupt Enabled	1
14	R/W	PCIE_RADMX_CMP SR_LU_ERR_E6	Controls reporting (interrupt generation) of PCIe-AXI Bridge RADMX Composer Lookup Error. 0: Interrupt Disabled 1: Interrupt Enabled	1
13	R/W	PCIE_RADM_CPL_TI MEOUT_E6	Controls reporting (interrupt generation) of PCIe Core RADM Completion Timeout Error. 0: Interrupt Disabled 1: Interrupt Enabled	1
12	R/W	PCIE_TRGT_CPL_TI MEOUT_E6	Controls reporting (interrupt generation) of PCIe Core TRGT Completion Timeout Error. 0: Interrupt Disabled 1: Interrupt Enabled	1
11	R/W	PCIE_CFG_AER_RC _ERR_INT_E6	Controls reporting (interrupt generation) of Error that cause bits in the Root Error Status Register of PCIe Core. 0: Interrupt Disabled 1: Interrupt Enabled	1
10	R/W	PCIE_CFG_AER_RC _ERR_MSI_E6	Controls reporting (interrupt generation) of Error that caused bits in the Root Error Status Register of PCIe Core when MSI or MSIX is enabled. 0: Interrupt Disabled 1: Interrupt Enabled	1
9	R/W	PCIE_CFG_SYS_ER R_RC_E6	Controls reporting (interrupt generation) when as a root complex any device in the hierarchy reports ERR_COR, NONFATAL_ERR, or FATAL_ERR. 0: Interrupt Disabled 1: Interrupt Enabled	1

Bit(s)	Access	Name	Description	Reset State
8	R/W	PCIE_RADM_CORRECTABLE_ERR_E6	Controls reporting (interrupt generation) of received of ERR_COR message. 0: Interrupt Disabled 1: Interrupt Enabled	1
7	R/W	PCIE_RADM_NONFATAL_ERR_E6	Controls reporting (interrupt generation) of a received NERR_ONFATAL message. 0: Interrupt Disabled 1: Interrupt Enabled	1
6	R/W	PCIE_RADM_FATAL_ERR_E6	Controls reporting (interrupt generation) of a received ERR_NONFATAL message. 0: Interrupt Disabled 1: Interrupt Enabled	1
5	R/W	PCIE_RETRY_RAM_PAR_ERR_E6	Controls reporting (interrupt generation) if a parity error is detected in the PCIe core Retry RAM. 0: Interrupt Disabled 1: Interrupt Enabled	1
4	R/W	PCIE_RETRY_SOT_PAR_ERR_E6	Controls reporting (interrupt generation) if a parity error is detected in the PCIe core SOT Retry RAM. 0: Interrupt Disabled 1: Interrupt Enabled	1
3	R/W	PCIE_HDRQ_PAR_ERR_E6	Controls reporting (interrupt generation) if a parity error is detected in the PCIe core Header Queue. 0: Interrupt Disabled 1: Interrupt Enabled	1
2	R/W	PCIE_DATAQ_PAR_ERR_E6	Controls reporting (interrupt generation) if a parity error is detected in the PCIe core Data Queue. 0: Interrupt Disabled 1: Interrupt Enabled	1
1	R/W	AL_OB_TRANS_ERR_OR_E6	Controls reporting (interrupt generation) if an outbound AXI transaction causes a fatal error. 0: Interrupt Disabled 1: Interrupt Enabled	1
0	R/W	AL_OB_WRID_MISS_E6	Controls reporting (interrupt generation) if an outbound AXI write transaction response does not have the proper ID. 0: Interrupt Disabled 1: Interrupt Enabled	1

10.3.11 PCIe Error Interrupt Register

Mnemonic: ERROR_INT_STAT

Address: (via MEMBASE-II): 0x304C

Description: This register maintains the interrupt event bits for the errors that can be generated by the PCIe Core and the PCIe application layer.

The interrupt bits are cleared by reading (if wcimode=0) or writing a '1' (if wcimode=1).

Table 242 PCIe Error Interrupt Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	AL_SLV_WR_DATA_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Slave Write Data Port.	0
30	R/W	AL_SLV_RD_DATA_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Slave Read Data Port.	0
29	R/W	AL_SLV_WR_ADDR_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Slave Write Address Port.	0
28	R/W	AL_SLV_RD_ADDR_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Slave Read Address Port.	0
27	R/W	AL_MSTR_WR_DATA_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Master Write Data Port.	0
26	R/W	AL_MSTR_RD_DATA_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Master Read Data Port.	0
25	R/W	AL_MSTR_WR_ADDR_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Master Write Address Port.	0
24	R/W	AL_MSTR_RD_ADDR_PAR_ERR_I6	Interrupt is asserted (if enabled) whenever a parity error is detected on the PCIE Application Layer AXI Master Read Address Port.	0

Bit(s)	Access	Name	Description	Reset State
23	R/W	AL_SLV_RD_RSP_SLV_ERR_I6	Interrupt is asserted (if enabled) whenever a SLV_ERR response is detected on the PCIE Application Layer AXI Slave Read Port.	0
22	R/W	AL_SLV_RD_RSP_DEC_ERR_I6	Interrupt is asserted (if enabled) whenever a DEC_ERR response is detected on the PCIE Application Layer AXI Slave Read Port.	0
21	R/W	AL_SLV_WR_RSP_SLV_ERR_I6	Interrupt is asserted (if enabled) whenever a SLV_ERR response is detected on the PCIE Application Layer AXI Slave Write Port.	0
20	R/W	AL_SLV_WR_RSP_DEC_ERR_I6	Interrupt is asserted (if enabled) whenever a DEC_ERR response is detected on the PCIE Application Layer AXI Slave Write Port.	0
19	R/W	AL_MSTR_RD_RSP_SLV_ERR_I6	Interrupt is asserted (if enabled) whenever a SLV_ERR response is detected on the PCIE Application Layer AXI Master Read Port.	0
18	R/W	AL_MSTR_RD_RSP_DEC_ERR_I6	Interrupt is asserted (if enabled) whenever a DEC_ERR response is detected on the PCIE Application Layer AXI Master Read Port. This interrupt will also be asserted in a read transaction misses all defined inbound windows or is filtered by the Security Filter Ranges.	0
17	R/W	AL_MSTR_WR_RSP_SLV_ERR_I6	Interrupt is asserted (if enabled) whenever a SLV_ERR response is detected on the PCIE Application Layer AXI Master Write Port.	0
16	R/W	AL_MSTR_WR_RSP_DEC_ERR_I6	Interrupt is asserted (if enabled) whenever a DEC_ERR response is detected on the PCIE Application Layer AXI Master Write Port. This interrupt will also be asserted in a write transaction misses all defined inbound windows or is filtered by the Security Filter Ranges.	0
15	R/W	PCIE_GM_CMPSR_LU_ERR_I6	Interrupt is asserted (if enabled) when PCIe-AXI Bridge Master Composer Lookup Error.	0
14	R/W	PCIE_RADMX_CMP_SR_LU_ERR_I6	Interrupt is asserted (if enabled) when PCIe-AXI Bridge RADMX Composer Lookup Error.	0
13	R/W	PCIE_RADM_CPL_TI_MEOUT_I6	Interrupt is asserted (if enabled) when PCIe Core RADM Completion Timeout Error.	0
12	R/W	PCIE_TRGT_CPL_TI_MEOUT_I6	Interrupt is asserted (if enabled) when PCIe Core TRGT Completion Timeout Error.	0

Bit(s)	Access	Name	Description	Reset State
11	R/W	PCIE_CFG_AER_RC_ERR_INT_I6	Interrupt is asserted (if enabled) when Error that cause bits in the Root Error Status Register of PCIe Core.	0
10	R/W	PCIE_CFG_AER_RC_ERR_MSI_I6	Interrupt is asserted (if enabled) when Error that caused bits in the Root Error Status Register of PCIe Core when MSI or MSIX is enabled.	0
9	R/W	PCIE_CFG_SYS_ER_R_RC_I6	Interrupt is asserted (if enabled) when as a root complex any device in the hierarchy reports ERR_COR, ERR_NONFATAL, or ERR_FATAL.	0
8	R/W	PCIE_RADM_CORR_ECTABLE_ERR_I6	Interrupt is asserted (if enabled) when an ERR_COR message is received.	0
7	R/W	PCIE_RADM_NONFATAL_ERR_I6	Interrupt is asserted (if enabled) when an ERR_NONFATAL message is received.	0
6	R/W	PCIE_RADM_FATAL_ERR_I6	Interrupt is asserted (if enabled) when an ERR_NONFATAL message is received.	0
5	R/W	PCIE_RETRY_RAM_PAR_ERR_I6	Interrupt is asserted (if enabled) when a parity error is detected in the PCIe core Retry RAM.	0
4	R/W	PCIE_RETRY_SOT_PAR_ERR_I6	Interrupt is asserted (if enabled) when a parity error is detected in the PCIe core SOT Retry RAM.	0
3	R/W	PCIE_HDRQ_PAR_ERR_I6	Interrupt is asserted (if enabled) when a parity error is detected in the PCIe core Header Queue.	0
2	R/W	PCIE_DATAQ_PAR_ERR_I6	Interrupt is asserted (if enabled) when a parity error is detected in the PCIe core Data Queue.	0
1	R/W	AL_OB_TRANS_ERR_OR_I6	Interrupt is asserted (if enabled) when an outbound AXI transaction causes a fatal error.	0
0	R/W	AL_OB_WRID_MISS_I6	Interrupt is asserted (if enabled) when an outbound AXI write transaction response does not have the proper ID.	0

10.3.12 Inbound AXI Translation Lower Address – Window 2 Register

Mnemonic: IB_AXI_ADDR_LO_2

Address: (via MEMBASE-II): 0x3258

Description:

Table 243 Inbound AXI Translation Lower Address – Window 2 Register Bits

Bit(s)	Access	Name	Description	Reset State
31:8	R/W	IB_AXI_ADDR_2 [31:8]	Programmed to the lower bits of the AXI Translation Address for inbound window 2.	0x000000
7:0	–	UNUSED		XXXX XXXX

10.3.13 Inbound AXI Translation Upper Address – Window 2 Register

Mnemonic: IB_AXI_ADDR_UP_2

Address: (via MEMBASE-II): 0x3268

Description:

Table 244 Inbound AXI Translation Lower Address – Window 2 Register Bits

Bit(s)	Access	Name	Description	Reset State
31:4	–	UNUSED		XXX XXXX XXXX XXXX XXXX XXXX XXXX
4:0	R/W	IB_AXI_ADDR_2 [36:32]	Programmed to the upper bits of the AXI Translation Address for inbound window 2.	10000

10.3.14 Message Unit Outbound Doorbell Auto Clear Register

Mnemonic: Message Unit Outbound Doorbell Auto Clear Register

Address: (via MEMBASE-II): 0x335C

Description: Usually set once during initialization. When set, a bit in this register will cause the corresponding bit in the Outbound Doorbell register to automatically clear after the corresponding interrupt is queued to the Outbound Transaction Block.

Note: Interrupt coalescing will not work with the auto-clearing feature of the [Outbound Doorbell Register](#). Hence, bits with the auto-clear feature set should not have the corresponding bit set in the [Interrupt Coalescing Control Register](#).

Table 245 MSGU Outbound Doorbell Auto Clear Register Bits

Bit(s)	Access	Name	Description	Reset State
31:0	R/W	MU_OB_DB_AUTO_CLR [31:0]	When set, a bit in this register will cause the corresponding bit in the Outbound Doorbell register to automatically clear after the corresponding interrupt is queued to the Outbound Transaction Block.	0x00000000

10.3.15 Interrupt Coalescing Timer Register

Mnemonic: Interrupt Coalescing Timer

Address: (via MEMBASE-II): 0x33C0

Description: This register is a 16-bit timer with a resolution of 1 μ s. This is the timer used for interrupt delay/coalescing in the PCIe Application Layer. Zero is not a valid value. A value of 1 in the register will cause the interrupts to be normal. A value greater than 1 will cause coalescing delays.

Table 246 Interrupt Coalescing Timer Register Bits

Bit(s)	Access	Name	Description	Reset State
31:16	—	UNUSED		XXXX XXXX XXXX XXXX
15:0	R	INT_COALESCING_TIMER [15:0]	This is the timer used for interrupt delay/coalescing in the PCIe Application Layer.	0x0000

10.3.16 Interrupt Coalescing Control Register

Mnemonic: Interrupt Coalescing Control

Address: (via MEMBASE-II): 0x33C4

Description: This 32-bit register corresponds to the each of the 32 potential interrupts. When a bit in this register is set, the corresponding interrupt will be delayed until the timer expires.

Note: Interrupt coalescing will not work with the auto-clearing feature of the [Outbound Doorbell Register](#). Hence, bits with the auto-clear feature set should not have the corresponding bit set in the [Interrupt Coalescing Control Register](#).

Table 247 Interrupt Coalescing Control Register Bits

Bit(s)	Access	Name	Description	Reset State
31:0	R/W	INT_COALESCING_CTRL [31:0]	When a bit in this register is set, the corresponding interrupt will be delayed until the timer expires.	0x00000000

10.4 PCIe Gen2 PCS/PMA Module Registers

The 64-bit BAR specified at PCI configuration address 0x18 (MEMBASE-II) contains the following chip registers used for configuring the SPC 8x6G PCS/PMA module. The following subsections provide the descriptions for these registers.

Table 248 PCIe Gen2 PCS/PMA Module Address Map

Offset	MEMBASE	Name
0x400C	MEMBASE-II	Test Control/Status Register
0x4100 + 0x100*N(N=0:7)	MEMBASE-II	Lane N TRS Control Register
0x4108 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Transmitter Mode Register
0x410C + 0x100*N(N=0:7)	MEMBASE-II	Lane N Transmitter Control Register
0x4114 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Receiver RX Mode 1 Register
0x4118 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Receiver RX Mode 2 Register
0x4124 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Receiver Status Register
0x41B0 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Error Counter Control Register
0x41B4 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Disparity Error Count Register
0x41B8 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Code Violation Error Count Register
0x41BC + 0x100*N(N=0:7)	MEMBASE-II	Lane N PRBS and Test Pattern Error Count Register
0x41D0 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Diagnostics Configuration Register
0x41D4 + 0x100*N(N=0:7)	MEMBASE-II	Lane N PRBS Initial Value Register
0x41D8 + 0x100*N(N=0:7)	MEMBASE-II	Lane N PRBS Error Insertion Register
0x41DC + 0x100*N(N=0:7)	MEMBASE-II	Lane N Test Pattern Insertion Word 1_1 Register
0x41E0 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Test Pattern Insertion Word 1_2 Register
0x41E4 + 0x100*N(N=0:7)	MEMBASE-II	Lane N Test Pattern Insertion Word 2_1 Register
0x41EC + 0x100*N(N=0:7)	MEMBASE-II	Lane N Test Pattern Insertion Word 2_2 Register

10.4.1 Test Control/Status Register

Mnemonic: TEST_CTRL

Address: (via MEMBASE-II): 0x400C

Description:

Table 249 Test Control/Status Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:17]	—	UNUSED		XXX XXXX XXXX XXXX
16	R	DATA_RATE_STATUS	This bit indicates the data rate the PCIE SERDES are running at. 1: Serial data rate is 5.0 GHz. 0: Serial data rate is 2.5 GHz. Note that if DATA_RATE_OVR is set, then this bit should reflect the DATA_RATE_CTRL bit.	0
[15:3]	—	UNUSED		X XXXX XXXX XXXX
2	R/W	DATA_RATE_CTRL	If DATA_RATE_OVR is set, then this bit controls the rate the PCIE SERDES are running at. 1: Serial data rate is 5.0 GHz. 0: Serial data rate is 2.5 GHz. If DATA_RATE_OVR is clear, then this bit has no affect.	0
1	R/W	DATA_RATE_OVR	This bit provides the ability to override the data rate initiated through the link training process. 1: Serial data rate is specified by the DATA_RATE_CTRL bit. 0: Serial data rate is specified by the PCIe LTSSM state machine.	0
0	R/W	MASTER_TX_TEST_ENABLE	Provides a means by which all test patterns can be started simultaneously. If this bit is high, each lane starts transmitting test data when its respective Ln_PRBS_TX_EN bit is set. If it is desirous to start all lanes simultaneously, then this bit must be set low, then all the Ln_PRBS_TX_EN set high, then this bit set high.	1

10.4.2 Lane N TRS Control Register

Mnemonic: TRS0_CTRL

Address: (via MEMBASE-II): 0x4100 + 0x100*N(N=0:7)

Description:

Table 250 Lane 0 TRS Control Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	TRS0_ENB		0
30	R/W	RESERVED		0
29	R/W	RESERVED		1
28	R/W	RESERVED		0
27	R/W	TRS0_LBSEL_REG_OVRD	This bit set causes the TRS0_LBSEL input to the MABC to be controlled by the TRS0_LBSEL register bits.	0
[26:24]	R/W	TRS0_LBSEL	<p>TRS0_LBSEL specifies the TRS0_LBSEL input to the MABC when in loop back mode, or when TRS0_LBSEL_REG_OVRD is set.</p> <p>TRS0_LBSEL Mode</p> <ul style="list-style-type: none"> 000 Normal reception and transmission 001 Metallic loopback from DFE input to transmitter input 010 Metallic loopback from DCRU input to transmitter input 011 DCRU recovered clock to transmitter input 100 Diagnostic loopback from transmitter to receiver 101 CSU clock to transmitter output (divide-by-2) 110 Reserved 111 Reserved 	100
23	R/W	TRS0_PREEN_50G	TRS0_PREEN_50G enables pre-emphasis when the serial link is operating at 5.0 GHz. Setting this bit to 1 enabled pre-emphasis. Setting this bit to 0 disables pre-emphasis.	1
22	R/W	TRS0_PREEN_25G	TRS0_PREEN_25G enables pre-emphasis when the serial link is operating at 2.5 GHz. Setting this bit to 1 enabled pre-emphasis. Setting this bit to 0 disables pre-emphasis.	1
21	R/W	RESERVED		0
20	R/W	RESERVED		0
[19:18]	R/W	RESERVED		00
[17:16]	R/W	RESERVED		00

Bit(s)	Access	Name	Description	Reset State
[15:0]	R/W	TRS0_MODE	These bits specify the TRS0_MODE inputs to the SERDES. If TRS0_MODE[13]=0, function is disabled (default)	0000 0000 0000 0001

10.4.3 Lane N Transmitter Mode Register

Mnemonic: T0_MODE

Address: (via MEMBASE-II): 0x4108 + 0x100*N(N=0:7)

Description:

Table 251 Lane 0 Transmitter Mode Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	RESERVED		0
30	R/W	RESERVED		0
[29:28]	R/W	RESERVED		01
[27:26]	R/W	RESERVED		00
[25:22]	R/W	T0_50G_TX_PRE_EMP_3_0	5.0 Gbps low level output swing when pre-emphasis is enabled via TRS0_PREEN_50G bit in the Lane N TRS Control Register. TX_PRE_EMP_3_0 Pre- Emphasis (dB) 0000 0 0001 0.8 0010 1.6 0011 2.5 0100 3.5 0101 4.7 0110 6.0 0111 7.6 1000 9.5 1001 12.0 1010 15.6 1011 21.6 1100 infinity (squench)	0110

Bit(s)	Access	Name	Description	Reset State
[21:18]	R/W	T0_50G_TX_SWING_3_0	5.0 Gbps output swing levels when there is no pre-emphasis. TX_SWING_3_0 Nominal_Swing (mVppd) 0000 430 0001 480 0010 530 0011 580 0100 620 0101 670 0110 720 0111 770 1000 820 1001 860 1010 910 1011 1010 1100 1100 1101 1200	1011
[17:16]	R/W	T0_50G_EDGE_RATE_1_0	For 5.0 Gbps operation, controls the edge rate for the transmit signal. EDGE_RATE_1_0 Edge Rate (ps) 00 68 01 80 10 91 11 115	01
15	R/W	RESERVED		0
14	R/W	RESERVED		0
[13:12]	R/W	RESERVED		01
[11:10]	R/W	RESERVED		00

Bit(s)	Access	Name	Description	Reset State																														
[9:6]	R/W	T0_25G_TX_PRE_EMP_3_0	<p>2.5 Gbps low level output swing when pre-emphasis is enabled via TRS0_PREEN_25G bit in the Lane N TRS Control Register.</p> <table> <tr><td>TX_PRE_EMP_3_0</td><td>Pre-Emphasis (dB)</td></tr> <tr><td>0000</td><td>0</td></tr> <tr><td>0001</td><td>0.8</td></tr> <tr><td>0010</td><td>1.6</td></tr> <tr><td>0011</td><td>2.5</td></tr> <tr><td>0100</td><td>3.5</td></tr> <tr><td>0101</td><td>4.7</td></tr> <tr><td>0110</td><td>6.0</td></tr> <tr><td>0111</td><td>7.6</td></tr> <tr><td>1000</td><td>9.5</td></tr> <tr><td>1001</td><td>12.0</td></tr> <tr><td>1010</td><td>15.6</td></tr> <tr><td>1011</td><td>12.6</td></tr> <tr><td>1100</td><td>Infinity (squench)</td></tr> </table>	TX_PRE_EMP_3_0	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	12.6	1100	Infinity (squench)	0101		
TX_PRE_EMP_3_0	Pre-Emphasis (dB)																																	
0000	0																																	
0001	0.8																																	
0010	1.6																																	
0011	2.5																																	
0100	3.5																																	
0101	4.7																																	
0110	6.0																																	
0111	7.6																																	
1000	9.5																																	
1001	12.0																																	
1010	15.6																																	
1011	12.6																																	
1100	Infinity (squench)																																	
[5:2]	R/W	T0_25G_TX_SWING_3_0	<p>2.5 Gbps output swing levels when there is no pre-emphasis.</p> <table> <tr><td>TX_SWING_3_0</td><td>Nominal_Swing (mV/ppd)</td></tr> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> </table>	TX_SWING_3_0	Nominal_Swing (mV/ppd)	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1011
TX_SWING_3_0	Nominal_Swing (mV/ppd)																																	
0000	430																																	
0001	480																																	
0010	530																																	
0011	580																																	
0100	620																																	
0101	670																																	
0110	720																																	
0111	770																																	
1000	820																																	
1001	860																																	
1010	910																																	
1011	1010																																	
1100	1100																																	
1101	1200																																	

Bit(s)	Access	Name	Description	Reset State
[1:0]	R/W	T0_25G_EDGE_RATE_1_0	For 2.5 Gbps operation, controls the edge rate for the transmit signal. EDGE_RATE_1_0 Edge Rate (ps) 00 68 01 80 10 91 11 115	10

10.4.4 Lane N Transmitter Control Register

Mnemonic: T0_CTRL

Address: (via MEMBASE-II): 0x410C + 0x100*N(N=0:7)

Description:

Table 252 Lane 0 Transmitter Control Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:25]	—	UNUSED		XXX XXXX
24	R/W	RESERVED		0
[23:17]	—	UNUSED		XXX XXXX
16	R/W	RESERVED		1
[15:14]	R/W	RESERVED		00
[13:12]	R/W	T0_CTRL_50G	T0_CTRL_50G[5:4] are used to control the termination resistors when the serial link is operating at 5.0 GHz. When T0_50G_TX_SWING_3_0 bits in the Lane N Transmitter Mode Register are set to 1100 or 1101, T0_CTRL_50G must be set to 01.	00
[11:8]	R/W	RESERVED		0000
[7:6]	R/W	RESERVED		00
[5:4]	R/W	T0_CTRL_25G	T0_CTRL_25G[5:4] are used to control the termination resistors when the serial link is operating at 2.5 GHz. When T0_25G_TX_SWING_3_0 bits in the Lane N Transmitter Mode Register are set to 1100 or 1101, T0_CTRL_50G must be set to 01.	00
[3:0]	R/W	RESERVED		0000

10.4.5 Lane N Receiver RX Mode 1 Register

Mnemonic: R0_RX_MODE_1

Address: (via MEMBASE-II): 0x4114 + 0x100*N(N=0:7)

Description:

Table 253 Lane 0 Receiver RX Mode 1 Register Bits

Bit(s)	Access	Name	Description	Reset State																												
[31:21]	—	UNUSED		XXX XXXX XXXX																												
[20:15]	R/W	RESERVED		0 0000																												
[14:5]	R/W	R0_RXMODE_25G_14_5	<p>R0_RXMODE_25G_14_5 bits set the receiver equalizer peaking. See Table 12 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R0_RXMODE_25G_14_5</th> </tr> </thead> <tbody> <tr><td>0</td><td>11 1011 1010</td></tr> <tr><td>1</td><td>11 1001 0111</td></tr> <tr><td>2</td><td>11 1001 0110</td></tr> <tr><td>3</td><td>11 1001 0101</td></tr> <tr><td>4</td><td>11 1001 0100</td></tr> <tr><td>5</td><td>11 1001 0011</td></tr> <tr><td>6</td><td>11 1001 0010</td></tr> <tr><td>7</td><td>11 1001 0001</td></tr> <tr><td>8</td><td>11 1001 0000</td></tr> <tr><td>9</td><td>01 0011 0000</td></tr> <tr><td>10</td><td>01 0010 0000</td></tr> <tr><td>11</td><td>01 0001 0000</td></tr> <tr><td>12</td><td>01 0000 0000</td></tr> </tbody> </table>	Level	R0_RXMODE_25G_14_5	0	11 1011 1010	1	11 1001 0111	2	11 1001 0110	3	11 1001 0101	4	11 1001 0100	5	11 1001 0011	6	11 1001 0010	7	11 1001 0001	8	11 1001 0000	9	01 0011 0000	10	01 0010 0000	11	01 0001 0000	12	01 0000 0000	11 1011 1010
Level	R0_RXMODE_25G_14_5																															
0	11 1011 1010																															
1	11 1001 0111																															
2	11 1001 0110																															
3	11 1001 0101																															
4	11 1001 0100																															
5	11 1001 0011																															
6	11 1001 0010																															
7	11 1001 0001																															
8	11 1001 0000																															
9	01 0011 0000																															
10	01 0010 0000																															
11	01 0001 0000																															
12	01 0000 0000																															
[4:0]	R/W	RESERVED		1 1011																												

10.4.6 Lane N Receiver RX Mode 2 Register

Mnemonic: R0_RX_MODE_2

Address: (via MEMBASE-II): 0x4118 + 0x100*N(N=0:7)

Description:

Table 254 Lane 0 Receiver RX Mode 2 Register Bits

Bit(s)	Access	Name	Description	Reset State																										
[31:21]	—	UNUSED		XXX XXXX XXXX																										
[20:15]	R/W	RESERVED		0 0000																										
[14:5]	R/W	R0_RXMODE_50G_14_5	<p>R0_RXMODE_50G_14_5 bits set the receiver equalizer peaking. See Table 13 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R0_RXMODE_25G_14_5</th> </tr> </thead> <tbody> <tr><td>0</td><td>11 1010 1010</td></tr> <tr><td>1</td><td>11 0101 0101</td></tr> <tr><td>2</td><td>11 0101 0100</td></tr> <tr><td>3</td><td>11 0101 0011</td></tr> <tr><td>4</td><td>11 0101 0010</td></tr> <tr><td>5</td><td>11 0101 0001</td></tr> <tr><td>6</td><td>11 0101 0000</td></tr> <tr><td>7</td><td>11 0100 0000</td></tr> <tr><td>8</td><td>01 0011 0000</td></tr> <tr><td>9</td><td>01 0010 0000</td></tr> <tr><td>10</td><td>01 0001 0000</td></tr> <tr><td>11</td><td>01 0000 0000</td></tr> </tbody> </table>	Level	R0_RXMODE_25G_14_5	0	11 1010 1010	1	11 0101 0101	2	11 0101 0100	3	11 0101 0011	4	11 0101 0010	5	11 0101 0001	6	11 0101 0000	7	11 0100 0000	8	01 0011 0000	9	01 0010 0000	10	01 0001 0000	11	01 0000 0000	11 1010 1010
Level	R0_RXMODE_25G_14_5																													
0	11 1010 1010																													
1	11 0101 0101																													
2	11 0101 0100																													
3	11 0101 0011																													
4	11 0101 0010																													
5	11 0101 0001																													
6	11 0101 0000																													
7	11 0100 0000																													
8	01 0011 0000																													
9	01 0010 0000																													
10	01 0001 0000																													
11	01 0000 0000																													
[4:0]	R/W	RESERVED		1 1011																										

10.4.7 Lane N Receiver Status Register

Mnemonic: R0_STAT

Address: (via MEMBASE-II): 0x4124 + 0x100*N(N=0:7)

Description:

Table 255 Lane 0 Receiver Status Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:1]	—	UNUSED		XXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
0	R	R0_RATE_DET	Indicates the level of the R0_RATE_DET output of the MABC.	0

10.4.8 Lane N Error Counter Control Register

Mnemonic: L0_ERR_CNT_CNTL

Address: (via MEMBASE-II): 0x41B0 + 0x100*N(N=0:7)

Description:

Table 256 Lane 0 Error Counter Control Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:2]	—	UNUSED		XX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
1	R/W	L0_LCLK_CLEAR	LCLK_CLEAR is the count transfer clear enable. When set to logic: 1: Causes the error counters to be cleared when the counter values are transferred to their holding registers (when logic 1 is written to the LCLK bit). 0: The error counters are not cleared when the counter values are transferred to their holding registers.	0

Bit(s)	Access	Name	Description	Reset State
0	R/W	L0_LCLK	<p>LCLK is the counter transfer clock.</p> <p>Writing this register bit to logic 1 when LCLK_CLEAR = '0' triggers the transfer of all counter values to their holding registers. This bit is cleared when LCLK asserts high.</p> <p>Writing this register bit to logic 1 when LCLK_CLEAR = '1' triggers the transfer of all counter values to their holding registers, and clears all counters. This bit is self-clearing.</p>	0

10.4.9 Lane N Disparity Error Count Register

Mnemonic: L0_DISP_ERR_CNT

Address: (via MEMBASE-II): 0x41B4 + 0x100*N(N=0:7)

Description:

Table 257 Lane 0 Disparity Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	—	UNUSED		XXXX XXXX XXXX XXXX
[15:0]	R	L0_DISP_ERR_CNT	<p>DISP_ERR_CNT[15:0] indicates the number of DWords containing running disparity errors that have been received outside of the PHY reset sequence.</p> <p>DISP_ERR_CNT[15:0] saturates at a value of all ones ($2^{16}-1$).</p> <p>This count is only updated after PHYRDY = '1'.</p> <p>When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset.</p> <p>LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000

10.4.10 Lane N Code Violation Error Count Register

Mnemonic: L0_CODE_ERR_CNT

Address: (via MEMBASE-II): 0x41B8 + 0x100*N(N=0:7)

Description:

Table 258 Lane 0 Code Violation Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	—	UNUSED		XXXX XXXX XXXX XXXX
[15:0]	R	L0_CODE_VIOL_ER_CNT	<p>CODE_VIOL_ERR_CNT[15:0] indicates the number of DWords containing 8B/10B code violation errors that have been received outside of the PHY reset sequence.</p> <p>CODE_VIOL_ERR_CNT[15:0] saturates at a value of all ones ($2^{16}-1$). This count is only updated after PHYRDY = '1'.</p> <p>When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset.</p> <p>LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000

10.4.11 Lane N PRBS and Test Pattern Error Count Register

Mnemonic: L0_PATT_ERR_CNT

Address: (via MEMBASE-II): 0x41BC + 0x100*N(N=0:7)

Description:

Table 259 Lane 0 PRBS and Test Pattern Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	R	L0_TEST_PAT_ERR_CNT	<p>TEST_PAT_ERR_CNT[15:0] indicates the number of bit errors that have been received by the Test Pattern checker in the receive data path.</p> <p>TEST_PAT_ERR_CNT[15:0] saturates at a value of all ones ($2^{16}-1$).</p> <p>When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset.</p> <p>LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000
[15:0]	R	L0_PRBS_ERR_CNT	<p>PRBS_ERR_CNT[15:0] indicates the number of bit errors that have been received by the PRBS checker in the receive data path.</p> <p>PHY_RST_FAILED_CNT[31:0] saturates at a value of all ones ($2^{16}-1$).</p> <p>When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset.</p> <p>LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000

10.4.12 Lane N Diagnostics Configuration Register

Mnemonic: L0_DIAG_CFG

Address: (via MEMBASE-II): 0x41D0 + 0x100*N(N=0:7)

Description:

Table 260 Lane 0 Diagnostics Configuration Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:11]	—	UNUSED		X XXXX XXXX XXXX XXXX XXXX
10	R/W	L0_PRBS_RX_SELECT	<p>This bit is used in debug mode to select the PRBS polynomial for the Tx pattern generator. When set to logic:</p> <p>1: PRBS-7 is enabled. 0: PRBS-23 is enabled.</p> <p>This bit is only valid when PRBS checking is enabled (PRBS_RX_EN = '1').</p>	0
9	R/W	L0_PRBS_TX_SELECT	<p>This bit is used in debug mode to select the PRBS polynomial for the Tx pattern generator. When set to logic:</p> <p>1: PRBS-7 is enabled. 0: PRBS-23 is enabled.</p> <p>This bit is only valid when PRBS transmission is enabled (PRBS_TX_EN = '1').</p>	0
8	R/W	L0_PRBS_ERR_INS_EN	<p>This bit is used in debug mode to insert a bit error into the transmit PRBS stream. This bit is self-clearing. The bit error must be set in the PRBS Error Insertion Mask Register (ERR_INS_MSK). When set to logic:</p> <p>1: PRBS bit error insertion is enabled. 0: PRBS bit error insertion is disabled.</p> <p>This bit is only valid when PRBS transmission is enabled (PRBS_TX_EN = '1')</p>	0

Bit(s)	Access	Name	Description	Reset State
7	R/W	L0_PRBS_INVERT	<p>This bit is used in debug mode to invert the transmit PRBS data stream. When set to logic:</p> <p>1: PRBS is inverted before it is transmitted.</p> <p>0: PRBS is not inverted before transmission.</p> <p>This bit is only valid when PRBS transmission is enabled (PRBS_TX_EN = '1')</p>	0
6	R/W	L0_PRBS_RX_EN	<p>This bit is used in debug mode to check PRBS in the receive data path. When set to logic:</p> <p>1: PRBS checking is enabled.</p> <p>0: PRBS checking is disabled.</p>	0
5	R/W	L0_PRBS_TX_EN	<p>This bit is used in debug mode to insert PRBS into the transmit data path. Each bit is bit mapped. When set to logic:</p> <p>1: PRBS insertion is enabled.</p> <p>0: PRBS insertion is disabled.</p>	0
4	R/W	L0_TEST_PAT_ERR_INS_EN	<p>This bit is used in debug mode to insert a bit error into the transmit Test Pattern stream. This bit is self-clearing. The bit error must be set in the Error Insertion Mask Register (ERR_INS_MSK). When set to logic:</p> <p>1: Test pattern bit error insertion is enabled.</p> <p>0: Test pattern bit error insertion is disabled.</p> <p>This bit is only valid when Test Pattern transmission is enabled (TEST_PAT_TX_EN = '1')</p>	0
3	R/W	L0_TEST_PAT_RX_EN	<p>This bit is used in debug mode to check test pattern insertion into the receive data path. When set to logic:</p> <p>1: Test pattern checking is enabled.</p> <p>0: Test pattern checking is disabled.</p>	0
2	R/W	L0_TEST_PAT_TX_EN	<p>This bit is used in debug mode to set test pattern insertion into the transmit data path. When set to logic:</p> <p>1: Test pattern insertion is enabled.</p> <p>0: Test pattern insertion is disabled.</p>	0

Bit(s)	Access	Name	Description	Reset State
1	R/W	L0_CODE_VIOL_ER_R_INS_EN	This bit is used in debug mode to insert a code violation error into the transmit data path. This bit is self-clearing. When set to logic: 1: Code violation error insertion is enabled. 0: Code violation error insertion is disabled.	0
0	R/W	L0_DISP_ERR_INS_EN	This bit is used in debug mode to insert a disparity error into the transmit data path. This bit is self-clearing. When set to logic: 1: Disparity error insertion is enabled. 0: Disparity error insertion is disabled.	0

10.4.13 Lane N PRBS Initial Value Register

Mnemonic: L0_PRBS_INIT_VAL

Address: (via MEMBASE-II): 0x41D4 + 0x100*N(N=0:7)

Description:

Table 261 Lane 0 PRBS Initial Value Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:23]	—	UNUSED		X XXXX XXXX
[22:0]	R/W	L0_PRBS_LFSR_LD_VAL	PRBS_LFSR_LD_VAL is the initial value of the transmit PRBS pattern generator.	111 1111 1111 1111 1111 1111

10.4.14 Lane N PRBS Error Insertion Register

Mnemonic: L0_ERR_INSERT

Address: (via MEMBASE-II): 0x41D8 + 0x100*N(N=0:7)

Description:

Table 262 Lane 0 PRBS Error Insertion Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:20]	—	UNUSED		XXXX XXXX XXXX
[19:10]	R/W	L0_ERR_INS_MSK_1	ERR_INS_MSK_1 is the 8B/10B-encoded byte 1 mask to indicate which bits in the transmit PRBS-7 or Test Pattern stream is in error when PRBS_ERR_INS_EN or TEST_PAT_ERR_INS_EN is set to logic 1.	00 0000 0000
[9:0]	R/W	L0_ERR_INS_MSK_0	ERR_INS_MSK_0 is the 8B/10B-encoded byte 0 (LSB) mask to indicate which bits in the transmit PRBS-7 or Test Pattern stream is in error when PRBS_ERR_INS_EN or TEST_PAT_ERR_INS_EN is set to logic 1.	00 0000 0000

10.4.15 Lane N Test Pattern Insertion Word 1_1 Register

Mnemonic: L0_TEST_PATT_WORD_1_1

Address: (via MEMBASE-II): 0x41DC + 0x100*N(N=0:7)

Description:

Table 263 Lane 0 Test Pattern Insertion Word 1_1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX
[29:20]	R/W	L0_TEST_PAT_INS_0_2	TEST_PAT_INS_0_2 is the 8B/10B-encoded byte 2 of the first test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_0_2 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

Bit(s)	Access	Name	Description	Reset State
[19:10]	R/W	L0_TEST_PAT_INS_0_1	TEST_PAT_INS_0_1 is the 8B/10B-encoded byte 1 of the first test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_0_1 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000
[9:0]	R/W	L0_TEST_PAT_INS_0_0	TEST_PAT_INS_0_0 is the 8B/10B-encoded byte 0 (LSB) of the first test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_0_0 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.4.16 Lane N Test Pattern Insertion Word 1_2 Register

Mnemonic: L0_TEST_PATT_WORD_1_2

Address: (via MEMBASE-II): 0x41E0 + 0x100*N(N=0:7)

Description:

Table 264 Lane 0 Test Pattern Insertion Word 1_2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:16]	R/W	L0_TEST_PAT_RPT_0	TEST_PAT_RPT_0 is the optional repeat count for the first test pattern DWord. When TEST_PAT_RPT_0 is non-zero, the first test pattern DWord is repeated an additional TEST_PAT_RPT_0 times each test pattern loop, for a total of (TEST_PAT_RPT_0 + 1) insertions of this DWord per loop. When TEST_PAT_RPT_0 is zero, the first test pattern DWord is inserted only once per test pattern loop.	0000 0000
[15:10]	—	UNUSED		XX XXXX

Bit(s)	Access	Name	Description	Reset State
[9:0]	R/W	L0_TEST_PAT_INS_0_3	TEST_PAT_INS_0_3 is the 8B/10B-encoded byte 3 (MSB) of the first test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_0_3 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.4.17 Lane N Test Pattern Insertion Word 2_1 Register

Mnemonic: L0_TEST_PATT_WORD_2_1

Address: (via MEMBASE-II): 0x41E4 + 0x100*N(N=0:7)

Description:

Table 265 Lane 0 Test Pattern Insertion Word 2_1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX
[29:20]	R/W	L0_TEST_PAT_INS_1_2	TEST_PAT_INS_1_2 is the 8B/10B-encoded byte 2 of the second test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_1_2 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000
[19:10]	R/W	L0_TEST_PAT_INS_1_1	TEST_PAT_INS_1_1 is the 8B/10B-encoded byte 1 of the second test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_1_1 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000
[9:0]	R/W	L0_TEST_PAT_INS_1_0	TEST_PAT_INS_1_0 is the 8B/10B-encoded byte 0 (LSB) of the second test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_1_0 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.4.18 Lane N Test Pattern Insertion Word 2_2 Register

Mnemonic: L0_TEST_PATT_WORD_2_2

Address: (via MEMBASE-II): 0x41EC + 0x100*N(N=0:7)

Description:

Table 266 Lane 0 Test Pattern Insertion Word 2_2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:16]	R/W	L0_TEST_PAT_RPT_1	TEST_PAT_RPT_1 is the optional repeat count for the second test pattern DWord. When TEST_PAT_RPT_1 is non-zero, the second test pattern DWord is repeated an additional TEST_PAT_RPT_1 times each test pattern loop, for a total of (TEST_PAT_RPT_1 + 1) insertions of this DWord per loop. When TEST_PAT_RPT_1 is zero, the second test pattern DWord is inserted only once per test pattern loop.	0000 0000
[15:10]	—	UNUSED		XX XXXX
[9:0]	R/W	L0_TEST_PAT_INS_1_3	TEST_PAT_INS_1_3 is the 8B/10B-encoded byte 3 (MSB) of the second test pattern DWord. When TEST_PAT_ERR_INS_EN is set to logic 1, TEST_PAT_INS_1_3 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.5 Octal SAS/SATA Port Subsystem Registers

The 32-bit BAR specified at PCI configuration address 0x20 (MEMBASE-III) contains the following chip registers used for configuring the SPC 8x6G OSSP subsystem. The following subsections provide the descriptions for these registers.

The host must perform MEMBASE-III inbound window-shifting aligned on 64 KB as shown in Section 2.6.1, “[MEMBASE-III Inbound Window Shifting](#)”. The remaining space provided in the offset must be used as the offset for reading/writing the register using MEMBASE-III.

Table 267 Octal SAS/SATA Port Subsystem Address Map

Offset	MEMBASE	Name
0x2_0000: 0x000	MEMBASE-III	OSSP Control 1 Register

Offset	MEMBASE	Name
0x2_0000: 0x008	MEMBASE-III	Timer Control 0 Register
0x2_0000: 0x010	MEMBASE-III	Timer Control 2 Register
0x2_0000: 0x100 + 0x100*N(N=0:7)	MEMBASE-III	Transmitter Per Port Configuration 1 SAS_SATA G1 Register
0x2_0000: 0x104 + 0x100*N(N=0:7)	MEMBASE-III	Transmitter Per Port Configuration 1 SAS_SATA G2 Register
0x2_0000: 0x108 + 0x100*N(N=0:7)	MEMBASE-III	Transmitter Per Port Configuration 1 SAS_SATA G3 Register
0x2_0000: 0x110 + 0x100*N(N=0:7)	MEMBASE-III	Receiver Per Port Configuration 1 SAS_SATA G1G2 Register
0x2_0000: 0x114 + 0x100*N(N=0:7)	MEMBASE-III	Receiver Per Port Configuration 1 SAS_SATA G3 Register
0x2_0000: 0x120 + 0x100*N(N=0:7)	MEMBASE-III	Global Configuration 1 Register
0x2_0000: 0x128 + 0x100*N(N=0:7)	MEMBASE-III	Transmitter Configuration 1 Register
0x2_0000: 0x130 + 0x100*N(N=0:7)	MEMBASE-III	Receiver Configuration 1 Register
0x2_0000: 0x134 + 0x100*N(N=0:7)	MEMBASE-III	Receiver Configuration 2 Register

10.5.1 OSSP Control 1 Register

Mnemonic: CONTROL_1

Address: (via MEMBASE-III):
Using shifted destination address 0x2_0000: 0x000

Description:

Table 268 OSSP Control 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:29]	—	Unused		XXX
28	R/W	SW_RESET	This bit resets the core logic of the OSSP. When set to logic '1', the SSPA core will be held in reset.	0
[27:24]	R/W	SAS_MUX_EN_MISALIGN_RST	When SAS_MUX_EN_MISALIGN_RST[n] is set to 1, the corresponding mux ports will reset on detection of misaligned MUX primitives. When set to logic 0, the ports will not reset on detection of misaligned MUX primitives, although they may reset for other reasons.	0000

Bit(s)	Access	Name	Description	Reset State
[23:20]	R/W	SAS_MUX_EN_ALIGN_RND	When SAS_MUX_EN_ALIGN_RND[n] is set to 1, the corresponding mux ports will re-randomize the ALIGNs after muxing. When set to logic 0, the ALIGNs are passed through the muxing layer transparently.	0000
[19:16]	R/W	SAS_MUX_ENABLE	When SAS_MUX_ENABLE[n] is set to 1, the muxing layer is enabled. Muxing will be allowed as follows: Muxed Ports SAS Enable Bit 0 and 4 SAS_MUX_ENABLE[0] 1 and 5 SAS_MUX_ENABLE[1] 2 and 6 SAS_MUX_ENABLE[2] 3 and 7 SAS_MUX_ENABLE[3]	0000
[15:14]	R/W	AXIM_RIO6_PRIOR	This register sets the read priority for AXI master port 6. These ports are used for writing to the GSM IO context.	00
[13:12]	R/W	AXIM_WIO6_PRIOR	This register sets the write priority for AXI master port 6. These ports are used for writing to the GSM IO context.	00
[11:10]	R/W	AXIM_RQ5_PRIOR	This register sets the read priority for AXI master port 5. These ports are used for writing to the GSM queue, which serves PHY port 4 to 7 of the OSSP.	00
[9:8]	R/W	AXIM_WQ5_PRIOR	This register sets the write priority for AXI master port 5. These ports are used for writing to the GSM queue, which serves PHY port 4 to 7 of the OSSP.	00
[7:6]	R/W	AXIM_RQ4_PRIOR	This register sets the read priority for AXI master port 4. These ports are used for writing to the GSM queue, which serves PHY port 4 to 7 of the OSSP.	00
[5:4]	R/W	AXIM_WQ4_PRIOR	This register sets the write priority for AXI master port 4. These ports are used for writing to the GSM queue, which serves PHY port 0 to 3 of the OSSP.	00
[3:2]	R/W	AXIM_RDATA03_PIOR	This register sets the read priority for AXI master port 0 to 3. These ports are used for data writing to the GSM.	00
[1:0]	R/W	AXIM_WDATA03_PIOR	This register sets the write priority for AXI master port 0 to 3. These ports are used for data writing to the GSM.	00

10.5.2 Timer Control 0 Register

Mnemonic: TIMER_0

Address: (via MEMBASE-III):
Using shifted destination address 0x2_0000: 0x008

Description:

Table 269 Timer Control 0 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	R/W	MAX_CONNECT_TIMER_SSP	This value is used for the maximum connection timeout timer in SSP connections. This value is in 100 µs units.	0000 0000 0010 0000
[15:0]	R/W	MAX_CONNECT_TIMER_STP	This value is used for the maximum connection timeout timer in STP connections. This value is in 100 µs unit.	0000 0000 0010 0000

10.5.3 Timer Control 2 Register

Mnemonic: TIMER_2

Address: (via MEMBASE-III):
Using shifted destination address 0x2_0000: 0x010

Description:

Table 270 Timer Control 2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:28]	—	Unused		XXXX
[27:24]	R/W	Reserved	These bits are for debug purposes. To ensure that the device operates as intended, the host must not change this value.	0000
[23:16]	R/W	PROTOCOL_TIMEOUT	This value is used for various internal SAS link layer protocol timeout timers. This value is in 20 µs units. The default is 1 ms. The host must not change this value.	0011 0010
[15:0]	R/W	STP_FRAME_TIMEOUT	This value is used for SATA/STP frame timeout timer. This value is in 1 µs units. The host may change this value (in 1 µs units) to alter the timeout value for the STP frame timer.	0000 0000 0000 0000

10.5.4 Transmitter Per Port Configuration 1 SAS_SATA G1 Register

Mnemonic: TX_PPC_SAS_SATA_G1

Address: (via MEMBASE-III):
Using shifted destination address 0x2_0000: 0x100 + 0x100*N(N=0:7)

Description:

Table 271 Transmitter Per Port Configuration 1 SAS_SATA G1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	R/W	T_MODE_13_12_SATA_G1	The T_MODE_13_12 bits control the transmitter bias. When external pull-up inductors/resistors are used, these bits must be set to 10. When no external pull-up inductors/resistors are used, these bits must be set to 01.	10
29	R/W	TRS_SSCEN_SATA_G1	Spread-spectrum enable. 0 = SSC disabled 1 = SSC enabled	0
28	R/W	TRS_PREEN_SATA_G1	Transmit Pre-emphasis enable 0 = Pre-emphasis disabled 1 = Pre-emphasis enabled	0
[27:26]	R/W	T_CTRL_7_6_SATA_G1	For SATA 1.5 Gbps operation. Pre-driver bias current control. See T_MODE_15_SATA_G1 description in the Transmitter Configuration 1 Register for current settings. These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.	01

Bit(s)	Access	Name	Description	Reset State																												
[25:24]	R/W	T_CTRL_5_4_SATA_G1	<p>The T_CTRL_5_4_SATA_G1 bits control the Tx common mode shift for applications that do not use external pull-up inductors/resistors on the SAS Tx PHYs.</p> <p>The default value of 00 is used when external pull-up inductors/resistors are present.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1350 or 1500 mVppd are used, the T_CTRL_5_4 bits must be set to 11.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1200 or 1100 mVppd are used, T_CTRL_5_4 bits must be set to 01.</p> <p>All lower settings must use 00.</p>	00																												
[23:20]	R/W	T_MODE_9_6_SATA_G1	<p>T_MODE[9:6] controls the TX_1.5G low level output swing when pre-emphasis is enabled (PREEN=1) for SATA 1.5 Gbps operation.</p> <p>By default, it is recommended that T_MODE[9:6] be set to "0000", which is a valid pre-emphasis setting for all modes.</p> <table> <thead> <tr> <th>T_MODE[9:6]</th> <th>Pre-Emphasis (dB)</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>0</td> </tr> <tr> <td>0001</td> <td>0.8</td> </tr> <tr> <td>0010</td> <td>1.6</td> </tr> <tr> <td>0011</td> <td>2.5</td> </tr> <tr> <td>0100</td> <td>3.5</td> </tr> <tr> <td>0101</td> <td>4.7</td> </tr> <tr> <td>0110</td> <td>6.0</td> </tr> <tr> <td>0111</td> <td>7.6</td> </tr> <tr> <td>1000</td> <td>9.5</td> </tr> <tr> <td>1001</td> <td>12.0</td> </tr> <tr> <td>1010</td> <td>15.6</td> </tr> <tr> <td>1011</td> <td>21.6</td> </tr> <tr> <td>1100</td> <td>infinity (squench)</td> </tr> </tbody> </table> <p>T_MODE[9:6] selects the non-emphasized output level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[9:6]	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	21.6	1100	infinity (squench)	0000
T_MODE[9:6]	Pre-Emphasis (dB)																															
0000	0																															
0001	0.8																															
0010	1.6																															
0011	2.5																															
0100	3.5																															
0101	4.7																															
0110	6.0																															
0111	7.6																															
1000	9.5																															
1001	12.0																															
1010	15.6																															
1011	21.6																															
1100	infinity (squench)																															

Bit(s)	Access	Name	Description	Reset State																																		
[19:16]	R/W	T_MODE_5_2_SATA_G1	<p>T_MODE[5:2] controls the TX_1.5G output swing levels when there is no pre-emphasis (TRS_PREEN='0') for 1.5 Gbps operation.</p> <p>When pre-emphasis is enabled (TRS_PREEN='1'), T_MODE[5:2] sets the high level, while T_MODE[9:6] sets the low level.</p> <table> <thead> <tr> <th>T_MODE[5:2]</th> <th>Nominal_Swing (mVppd)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> <tr><td>1110</td><td>1350</td></tr> <tr><td>1111</td><td>1500</td></tr> </tbody> </table> <p>T_MODE[5:2] select the full output swing level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[5:2]	Nominal_Swing (mVppd)	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1110	1350	1111	1500	0010
T_MODE[5:2]	Nominal_Swing (mVppd)																																					
0000	430																																					
0001	480																																					
0010	530																																					
0011	580																																					
0100	620																																					
0101	670																																					
0110	720																																					
0111	770																																					
1000	820																																					
1001	860																																					
1010	910																																					
1011	1010																																					
1100	1100																																					
1101	1200																																					
1110	1350																																					
1111	1500																																					
[15:14]	R/W	T_MODE_13_12_SAS_G1	The T_MODE_13_12 bits control the transmitter bias. When external pull-up inductors/resistors are used, these bits must be set to 10. When no external pull-up inductors/resistors are used, these bits must be set to 01.	10																																		
13	R/W	TRS_SSCEN_SAS_G1	Spread-spectrum enable. 0: SSC disabled 1: SSC enabled	0																																		

Bit(s)	Access	Name	Description	Reset State
12	R/W	TRS_PREEN_SAS_G1	<p>Pre-emphasis enable in the PISO and transmitter.</p> <p>When set to logic:</p> <p>1: Transmit path pre-emphasis is enabled.</p> <p>0: Transmit path pre-emphasis is disabled.</p> <p>TRS_PREEN is controllable on a per-rate basis and a SAS/SATA basis.</p>	0
[11:10]	R/W	T_CTRL_7_6_SAS_G1	<p>For SAS 1.5 Gbps operation. Pre-driver bias current control.</p> <p>See T_MODE_15_SAS_G1 description the Transmitter Configuration 1 Register for current settings.</p> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	00
[9:8]	R/W	T_CTRL_5_4_SAS_G1	<p>The T_CTRL_5_4_SAS_G1 bits control the Tx common mode shift for applications that do not use external pull-up inductors/resistors on the SAS Tx PHYs.</p> <p>The default value of 00 is used when external pull-up inductors/resistors are present.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1350 or 1500 mVppd are used, the T_CTRL_5_4 bits must be set to 11.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1200 or 1100 mVppd are used, T_CTRL_5_4 bits must be set to 01.</p> <p>All lower settings must use 00.</p>	00

Bit(s)	Access	Name	Description	Reset State																												
[7:4]	R/W	T_MODE_9_6_SAS_G1	<p>T_MODE[9:6] controls the TX_1.5G low level output swing when pre-emphasis is enabled (PREEN=1) for 1.5 Gbps operation.</p> <p>By default, it is recommended that T_MODE[9:6] be set to "0000", which is a valid pre-emphasis setting for all modes.</p> <table> <thead> <tr> <th>T_MODE[9:6]</th> <th>Pre-Emphasis (dB)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>0</td></tr> <tr><td>0001</td><td>0.8</td></tr> <tr><td>0010</td><td>1.6</td></tr> <tr><td>0011</td><td>2.5</td></tr> <tr><td>0100</td><td>3.5</td></tr> <tr><td>0101</td><td>4.7</td></tr> <tr><td>0110</td><td>6.0</td></tr> <tr><td>0111</td><td>7.6</td></tr> <tr><td>1000</td><td>9.5</td></tr> <tr><td>1001</td><td>12.0</td></tr> <tr><td>1010</td><td>15.6</td></tr> <tr><td>1011</td><td>21.6</td></tr> <tr><td>1100</td><td>infinity (squench)</td></tr> </tbody> </table> <p>T_MODE[9:6] selects the non-emphasized output level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[9:6]	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	21.6	1100	infinity (squench)	0000
T_MODE[9:6]	Pre-Emphasis (dB)																															
0000	0																															
0001	0.8																															
0010	1.6																															
0011	2.5																															
0100	3.5																															
0101	4.7																															
0110	6.0																															
0111	7.6																															
1000	9.5																															
1001	12.0																															
1010	15.6																															
1011	21.6																															
1100	infinity (squench)																															

Bit(s)	Access	Name	Description	Reset State																																		
[3:0]	R/W	T_MODE_5_2_SAS_G1	<p>T_MODE[5:2] controls the TX_1.5G output swing levels when there is no pre-emphasis (TRS_PREEN='0') for 1.5 Gbps operation.</p> <p>When pre-emphasis is enabled (TRS_PREEN='1'), T_MODE[5:2] sets the high level, while T_MODE[9:6] sets the low level.</p> <table> <thead> <tr> <th>T_MODE[5:2]</th> <th>Nominal_Swing (mVppd)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> <tr><td>1110</td><td>1350</td></tr> <tr><td>1111</td><td>1500</td></tr> </tbody> </table> <p>T_MODE[5:2] select the full output swing level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[5:2]	Nominal_Swing (mVppd)	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1110	1350	1111	1500	1011
T_MODE[5:2]	Nominal_Swing (mVppd)																																					
0000	430																																					
0001	480																																					
0010	530																																					
0011	580																																					
0100	620																																					
0101	670																																					
0110	720																																					
0111	770																																					
1000	820																																					
1001	860																																					
1010	910																																					
1011	1010																																					
1100	1100																																					
1101	1200																																					
1110	1350																																					
1111	1500																																					

10.5.5 Transmitter Per Port Configuration 1 SAS_SATA G2 Register

Mnemonic: TX_PPC_SAS_SATA_G2

Address: (via MEMBASE-III):

Using shifted destination address 0x2_0000: 0x104 + 0x100*N(N=0:7)

Description:

Table 272 Transmitter Per Port Configuration 1 SAS_SATA G2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	R/W	T_MODE_13_12_SATA_G2	The T_MODE_13_12 bits control the transmitter bias. When external pull-up inductors/resistors are used, these bits must be set to 10. When no external pull-up inductors/resistors are used, these bits must be set to 01.	10
29	R/W	TRS_SSCEN_SATA_G2	Spread-spectrum enable. 0 = SSC disabled 1 = SSC enabled	0
28	R/W	TRS_PREEN_SATA_G2	Transmit Pre-emphasis enable 0 = Pre-emphasis disabled 1 = Pre-emphasis enabled	0
[27:26]	R/W	T_CTRL_7_6_SATA_G2	For SATA 3 Gbps operation. Pre-driver bias current control. See T_MODE_15_SATA_G2 description in the Transmitter Configuration 1 Register for current settings. These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.	01

Bit(s)	Access	Name	Description	Reset State																												
[25:24]	R/W	T_CTRL_5_4_SATA_G2	<p>The T_CTRL_5_4_SATA_G2 bits control the Tx common mode shift for applications that do not use external pull-up inductors/resistors on the SAS Tx PHYs.</p> <p>The default value of 00 is used when external pull-up inductors/resistors are present.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1350 or 1500 mVppd are used, the T_CTRL_5_4 bits must be set to 11.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1200 or 1100 mVppd are used, T_CTRL_5_4 bits must be set to 01.</p> <p>All lower settings must use 00.</p>	00																												
[23:20]	R/W	T_MODE_9_6_SATA_G2	<p>T_MODE[9:6] controls the TX_3G low level output swing when pre-emphasis is enabled (PREEN=1) for SATA 3 Gbps operation.</p> <p>By default, it is recommended that T_MODE[9:6] be set to "0000", which is a valid pre-emphasis setting for all modes.</p> <table> <thead> <tr> <th>T_MODE[9:6]</th> <th>Pre-Emphasis (dB)</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>0</td> </tr> <tr> <td>0001</td> <td>0.8</td> </tr> <tr> <td>0010</td> <td>1.6</td> </tr> <tr> <td>0011</td> <td>2.5</td> </tr> <tr> <td>0100</td> <td>3.5</td> </tr> <tr> <td>0101</td> <td>4.7</td> </tr> <tr> <td>0110</td> <td>6.0</td> </tr> <tr> <td>0111</td> <td>7.6</td> </tr> <tr> <td>1000</td> <td>9.5</td> </tr> <tr> <td>1001</td> <td>12.0</td> </tr> <tr> <td>1010</td> <td>15.6</td> </tr> <tr> <td>1011</td> <td>21.6</td> </tr> <tr> <td>1100</td> <td>infinity (squench)</td> </tr> </tbody> </table> <p>T_MODE[9:6] selects the non-emphasized output level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[9:6]	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	21.6	1100	infinity (squench)	0000
T_MODE[9:6]	Pre-Emphasis (dB)																															
0000	0																															
0001	0.8																															
0010	1.6																															
0011	2.5																															
0100	3.5																															
0101	4.7																															
0110	6.0																															
0111	7.6																															
1000	9.5																															
1001	12.0																															
1010	15.6																															
1011	21.6																															
1100	infinity (squench)																															

Bit(s)	Access	Name	Description	Reset State																																		
[19:16]	R/W	T_MODE_5_2_SAT_A_G2	<p>T_MODE[5:2] controls the TX_3G output swing levels when there is no pre-emphasis (TRS_PREEN='0') for 3 Gbps operation.</p> <p>When pre-emphasis is enabled (TRS_PREEN='1'), T_MODE[5:2] sets the high level, while T_MODE[9:6] sets the low level.</p> <table> <thead> <tr> <th>T_MODE[5:2]</th> <th>Nominal_Swing (mVppd)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> <tr><td>1110</td><td>1350</td></tr> <tr><td>1111</td><td>1500</td></tr> </tbody> </table> <p>T_MODE[5:2] select the full output swing level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[5:2]	Nominal_Swing (mVppd)	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1110	1350	1111	1500	0011
T_MODE[5:2]	Nominal_Swing (mVppd)																																					
0000	430																																					
0001	480																																					
0010	530																																					
0011	580																																					
0100	620																																					
0101	670																																					
0110	720																																					
0111	770																																					
1000	820																																					
1001	860																																					
1010	910																																					
1011	1010																																					
1100	1100																																					
1101	1200																																					
1110	1350																																					
1111	1500																																					
[15:14]	R/W	T_MODE_13_12_SAS_G2	The T_MODE_13_12 bits control the transmitter bias. When external pull-up inductors/resistors are used, these bits must be set to 10. When no external pull-up inductors/resistors are used, these bits must be set to 01.	10																																		
13	R/W	TRS_SSCEN_SAS_G2	Spread-spectrum enable. 0: SSC disabled 1: SSC enabled	0																																		

Bit(s)	Access	Name	Description	Reset State
12	R/W	TRS_PREEN_SAS_G2	<p>Pre-emphasis enable in the PISO and transmitter.</p> <p>When set to logic:</p> <ul style="list-style-type: none"> 1: Transmit path pre-emphasis is enabled. 0: Transmit path pre-emphasis is disabled. <p>TRS_PREEN is controllable on a per-rate basis and a SAS/SATA basis.</p>	0
[11:10]	R/W	T_CTRL_7_6_SAS_G2	<p>For SAS 3 Gbps operation. Pre-driver bias current control.</p> <p>See T_MODE_15_SAS_G2 description in the Transmitter Configuration 1 Register for current settings.</p> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	00
[9:8]	R/W	T_CTRL_5_4_SAS_G2	<p>The T_CTRL_5_4_SAS_G2 bits control the Tx common mode shift for applications that do not use external pull-up inductors/resistors on the SAS Tx PHYs.</p> <p>The default value of 00 is used when external pull-up inductors/resistors are present.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1350 or 1500 mVppd are used, the T_CTRL_5_4 bits must be set to 11.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1200 or 1100 mVppd are used, T_CTRL_5_4 bits must be set to 01.</p> <p>All lower settings must use 00.</p>	00

Bit(s)	Access	Name	Description	Reset State																												
[7:4]	R/W	T_MODE_9_6_SAS_G2	<p>T_MODE[9:6] controls the TX_3G low level output swing when pre-emphasis is enabled (PREEN=1) for 3 Gbps operation.</p> <p>By default, it is recommended that T_MODE[9:6] be set to "0000", which is a valid pre-emphasis setting for all modes.</p> <table> <thead> <tr> <th>T_MODE[9:6]</th> <th>Pre-Emphasis (dB)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>0</td></tr> <tr><td>0001</td><td>0.8</td></tr> <tr><td>0010</td><td>1.6</td></tr> <tr><td>0011</td><td>2.5</td></tr> <tr><td>0100</td><td>3.5</td></tr> <tr><td>0101</td><td>4.7</td></tr> <tr><td>0110</td><td>6.0</td></tr> <tr><td>0111</td><td>7.6</td></tr> <tr><td>1000</td><td>9.5</td></tr> <tr><td>1001</td><td>12.0</td></tr> <tr><td>1010</td><td>15.6</td></tr> <tr><td>1011</td><td>21.6</td></tr> <tr><td>1100</td><td>infinity (squench)</td></tr> </tbody> </table> <p>T_MODE[9:6] selects the non-emphasized output level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[9:6]	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	21.6	1100	infinity (squench)	0000
T_MODE[9:6]	Pre-Emphasis (dB)																															
0000	0																															
0001	0.8																															
0010	1.6																															
0011	2.5																															
0100	3.5																															
0101	4.7																															
0110	6.0																															
0111	7.6																															
1000	9.5																															
1001	12.0																															
1010	15.6																															
1011	21.6																															
1100	infinity (squench)																															

Bit(s)	Access	Name	Description	Reset State																																		
[3:0]	R/W	T_MODE_5_2_SAS_G2	<p>T_MODE[5:2] controls the TX_3G output swing levels when there is no pre-emphasis (TRS_PREEN='0') for 3 Gbps operation.</p> <p>When pre-emphasis is enabled (TRS_PREEN='1'), T_MODE[5:2] sets the high level, while T_MODE[9:6] sets the low level.</p> <table> <thead> <tr> <th>T_MODE[5:2]</th> <th>Nominal_Swing (mVppd)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> <tr><td>1110</td><td>1350</td></tr> <tr><td>1111</td><td>1500</td></tr> </tbody> </table> <p>T_MODE[5:2] select the full output swing level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[5:2]	Nominal_Swing (mVppd)	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1110	1350	1111	1500	1011
T_MODE[5:2]	Nominal_Swing (mVppd)																																					
0000	430																																					
0001	480																																					
0010	530																																					
0011	580																																					
0100	620																																					
0101	670																																					
0110	720																																					
0111	770																																					
1000	820																																					
1001	860																																					
1010	910																																					
1011	1010																																					
1100	1100																																					
1101	1200																																					
1110	1350																																					
1111	1500																																					

10.5.6 Transmitter Per Port Configuration 1 SAS_SATA G3 Register

Mnemonic: TX_PPC_SAS_SATA_G3

Address: (via MEMBASE-III):

Using shifted destination address 0x2_0000: 0x108 + 0x100*N(N=0:7)

Description:

Table 273 Transmitter Per Port Configuration 1 SAS_SATA G3 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	R/W	T_MODE_13_12_SATA_G3	The T_MODE_13_12 bits control the transmitter bias. When external pull-up inductors/resistors are used, these bits must be set to 10. When no external pull-up inductors/resistors are used, these bits must be set to 01.	10
29	R/W	TRS_SSCEN_SATA_G3	Spread-spectrum enable. 0 = SSC disabled 1 = SSC enabled	0
28	R/W	TRS_PREEN_SATA_G3	Transmit Pre-emphasis enable 0 = Pre-emphasis disabled 1 = Pre-emphasis enabled	0
[27:26]	R/W	T_CTRL_7_6_SATA_G3	For SATA 6 Gbps operation. Pre-driver bias current control. See T_MODE_15_SATA_G3 description in the Transmitter Configuration 1 Register for current settings. These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.	00

Bit(s)	Access	Name	Description	Reset State																												
[25:24]	R/W	T_CTRL_5_4_SATA_G3	<p>The T_CTRL_5_4_SATA_G3 bits control the Tx common mode shift for applications that do not use external pull-up inductors/resistors on the SAS Tx PHYs.</p> <p>The default value of 00 is used when external pull-up inductors/resistors are present.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1350 or 1500 mVppd are used, the T_CTRL_5_4 bits must be set to 11.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1200 or 1100 mVppd are used, T_CTRL_5_4 bits must be set to 01.</p> <p>All lower settings must use 00.</p>	00																												
[23:20]	R/W	T_MODE_9_6_SATA_G3	<p>T_MODE[9:6] controls the TX_6G low level output swing when pre-emphasis is enabled (PREEN=1) for 6 Gbps operation.</p> <p>By default, it is recommended that T_MODE[9:6] be set to "0000", which is a valid pre-emphasis setting for all modes.</p> <table> <thead> <tr> <th>T_MODE[9:6]</th> <th>Pre-Emphasis (dB)</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>0</td> </tr> <tr> <td>0001</td> <td>0.8</td> </tr> <tr> <td>0010</td> <td>1.6</td> </tr> <tr> <td>0011</td> <td>2.5</td> </tr> <tr> <td>0100</td> <td>3.5</td> </tr> <tr> <td>0101</td> <td>4.7</td> </tr> <tr> <td>0110</td> <td>6.0</td> </tr> <tr> <td>0111</td> <td>7.6</td> </tr> <tr> <td>1000</td> <td>9.5</td> </tr> <tr> <td>1001</td> <td>12.0</td> </tr> <tr> <td>1010</td> <td>15.6</td> </tr> <tr> <td>1011</td> <td>21.6</td> </tr> <tr> <td>1100</td> <td>infinity (squench)</td> </tr> </tbody> </table> <p>T_MODE[9:6] selects the non-emphasized output level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[9:6]	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	21.6	1100	infinity (squench)	0000
T_MODE[9:6]	Pre-Emphasis (dB)																															
0000	0																															
0001	0.8																															
0010	1.6																															
0011	2.5																															
0100	3.5																															
0101	4.7																															
0110	6.0																															
0111	7.6																															
1000	9.5																															
1001	12.0																															
1010	15.6																															
1011	21.6																															
1100	infinity (squench)																															

Bit(s)	Access	Name	Description	Reset State																																		
[19:16]	R/W	T_MODE_5_2_SATA_G3	<p>T_MODE[5:2] controls the TX_6G output swing levels when there is no pre-emphasis (TRS_PREEN='0') for 6 Gbps operation.</p> <p>When pre-emphasis is enabled (TRS_PREEN='1'), T_MODE[5:2] sets the high level, while T_MODE[9:6] sets the low level.</p> <table> <thead> <tr> <th>T_MODE[5:2]</th> <th>Nominal_Swing (mVppd)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> <tr><td>1110</td><td>1350</td></tr> <tr><td>1111</td><td>1500</td></tr> </tbody> </table> <p>T_MODE[5:2] select the full output swing level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[5:2]	Nominal_Swing (mVppd)	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1110	1350	1111	1500	1011
T_MODE[5:2]	Nominal_Swing (mVppd)																																					
0000	430																																					
0001	480																																					
0010	530																																					
0011	580																																					
0100	620																																					
0101	670																																					
0110	720																																					
0111	770																																					
1000	820																																					
1001	860																																					
1010	910																																					
1011	1010																																					
1100	1100																																					
1101	1200																																					
1110	1350																																					
1111	1500																																					
[15:14]	R/W	T_MODE_13_12_SAS_G3	The T_MODE_13_12 bits control the transmitter bias. When external pull-up inductors/resistors are used, these bits must be set to 10. When no external pull-up inductors/resistors are used, these bits must be set to 01.	10																																		
13	R/W	TRS_SSCEN_SAS_G3	Spread-spectrum enable. 0: SSC disabled 1: SSC enabled	0																																		

Bit(s)	Access	Name	Description	Reset State
12	R/W	TRS_PREEN_SAS_G3	<p>Pre-emphasis enable in the PISO and transmitter.</p> <p>When set to logic:</p> <p>1: Transmit path pre-emphasis is enabled.</p> <p>0: Transmit path pre-emphasis is disabled.</p>	1
[11:10]	R/W	T_CTRL_7_6_SAS_G3	<p>For SAS 6 Gbps operation. Pre-driver bias current control.</p> <p>See T_MODE_15_SAS_G3 description in the Transmitter Configuration 1 Register for current settings</p> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	00
[9:8]	R/W	T_CTRL_5_4_SAS_G3	<p>The T_CTRL_5_4_SAS_G3 bits control the Tx common mode shift for applications that do not use external pull-up inductors/resistors on the SAS Tx PHYs.</p> <p>The default value of 00 is used when external pull-up inductors/resistors are present.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1350 or 1500 mVppd are used, the T_CTRL_5_4 bits must be set to 11.</p> <p>When no external pull-up inductors/resistors are present and Tx swing levels of 1200 or 1100 mVppd are used, T_CTRL_5_4 bits must be set to 01.</p> <p>All lower settings must use 00.</p>	00

Bit(s)	Access	Name	Description	Reset State																												
[7:4]	R/W	T_MODE_9_6_SAS_G3	<p>T_MODE[9:6] controls the TX_6G low level output swing when pre-emphasis is enabled (PREEN=1) for 6 Gbps operation.</p> <p>By default, it is recommended that T_MODE[9:6] be set to "0000", which is a valid pre-emphasis setting for all modes.</p> <table> <thead> <tr> <th>T_MODE[9:6]</th> <th>Pre-Emphasis (dB)</th> </tr> </thead> <tbody> <tr><td>0000</td><td>0</td></tr> <tr><td>0001</td><td>0.8</td></tr> <tr><td>0010</td><td>1.6</td></tr> <tr><td>0011</td><td>2.5</td></tr> <tr><td>0100</td><td>3.5</td></tr> <tr><td>0101</td><td>4.7</td></tr> <tr><td>0110</td><td>6.0</td></tr> <tr><td>0111</td><td>7.6</td></tr> <tr><td>1000</td><td>9.5</td></tr> <tr><td>1001</td><td>12.0</td></tr> <tr><td>1010</td><td>15.6</td></tr> <tr><td>1011</td><td>21.6</td></tr> <tr><td>1100</td><td>infinity (squench)</td></tr> </tbody> </table> <p>T_MODE[9:6] selects the non-emphasized output level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	T_MODE[9:6]	Pre-Emphasis (dB)	0000	0	0001	0.8	0010	1.6	0011	2.5	0100	3.5	0101	4.7	0110	6.0	0111	7.6	1000	9.5	1001	12.0	1010	15.6	1011	21.6	1100	infinity (squench)	0101
T_MODE[9:6]	Pre-Emphasis (dB)																															
0000	0																															
0001	0.8																															
0010	1.6																															
0011	2.5																															
0100	3.5																															
0101	4.7																															
0110	6.0																															
0111	7.6																															
1000	9.5																															
1001	12.0																															
1010	15.6																															
1011	21.6																															
1100	infinity (squench)																															

Bit(s)	Access	Name	Description	Reset State																																
[3:0]	R/W	T_MODE_5_2_SAS_G3	<p>T_MODE[5:2] controls the TX_6G output swing levels when there is no pre-emphasis (TRS_PREEN='0') for 6 Gbps operation.</p> <p>When pre-emphasis is enabled (TRS_PREEN='1'), T_MODE[5:2] sets the high level, while T_MODE[9:6] sets the low level. T_MODE[5:2] Nominal_Swing (mVppd)</p> <table> <tbody> <tr><td>0000</td><td>430</td></tr> <tr><td>0001</td><td>480</td></tr> <tr><td>0010</td><td>530</td></tr> <tr><td>0011</td><td>580</td></tr> <tr><td>0100</td><td>620</td></tr> <tr><td>0101</td><td>670</td></tr> <tr><td>0110</td><td>720</td></tr> <tr><td>0111</td><td>770</td></tr> <tr><td>1000</td><td>820</td></tr> <tr><td>1001</td><td>860</td></tr> <tr><td>1010</td><td>910</td></tr> <tr><td>1011</td><td>1010</td></tr> <tr><td>1100</td><td>1100</td></tr> <tr><td>1101</td><td>1200</td></tr> <tr><td>1110</td><td>1350</td></tr> <tr><td>1111</td><td>1500</td></tr> </tbody> </table> <p>T_MODE[5:2] select the full output swing level and is controllable on a per-rate basis and on a SAS/SATA basis.</p>	0000	430	0001	480	0010	530	0011	580	0100	620	0101	670	0110	720	0111	770	1000	820	1001	860	1010	910	1011	1010	1100	1100	1101	1200	1110	1350	1111	1500	1101
0000	430																																			
0001	480																																			
0010	530																																			
0011	580																																			
0100	620																																			
0101	670																																			
0110	720																																			
0111	770																																			
1000	820																																			
1001	860																																			
1010	910																																			
1011	1010																																			
1100	1100																																			
1101	1200																																			
1110	1350																																			
1111	1500																																			

10.5.7 Receiver Per Port Configuration 1 SAS_SATA G1G2 Register

Mnemonic: RX_PPC_SAS_SATA_G1G2

Address: (via MEMBASE-III):
Using shifted destination address 0x2_0000: 0x110 + 0x100*N(N=0:7)

Description:

Table 274 Receiver Per Port Configuration 1 SAS_SATA G1G2 Register Bits

Bit(s)	Access	Name	Description	Reset State																												
[31:29]	R/W	RESERVED		001																												
28	R/W	RESERVED		1																												
27	R/W	RESERVED		1																												
[26:25]	R/W	R_RXMODE_14_13_SATA_G1G2	The RXMODE_14/13 Bit controls the receiver linear equalization in conjunction with the RXMODE_12_5 bits. Refer to the Peaking Mode Gain tables in the SPC 8x6G Hardware Specification for valid settings.	11																												
[24:21]	R/W	R_RXMODE_12_9_SATA_G1G2	R_RXMODE[12:9] controls receiver equalizer peaking frequency. See Table 12 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels. <table> <thead> <tr> <th>Level</th><th>R_RXMODE_12_9</th></tr> </thead> <tbody> <tr><td>0</td><td>1011</td></tr> <tr><td>1</td><td>1001</td></tr> <tr><td>2</td><td>1001</td></tr> <tr><td>3</td><td>1001</td></tr> <tr><td>4</td><td>1001</td></tr> <tr><td>5</td><td>1001</td></tr> <tr><td>6</td><td>1001</td></tr> <tr><td>7</td><td>1001</td></tr> <tr><td>8</td><td>1001</td></tr> <tr><td>9</td><td>0011</td></tr> <tr><td>10</td><td>0010</td></tr> <tr><td>11</td><td>0001</td></tr> <tr><td>12</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_12_9	0	1011	1	1001	2	1001	3	1001	4	1001	5	1001	6	1001	7	1001	8	1001	9	0011	10	0010	11	0001	12	0000	1011
Level	R_RXMODE_12_9																															
0	1011																															
1	1001																															
2	1001																															
3	1001																															
4	1001																															
5	1001																															
6	1001																															
7	1001																															
8	1001																															
9	0011																															
10	0010																															
11	0001																															
12	0000																															

Bit(s)	Access	Name	Description	Reset State																												
[20:17]	R/W	R_RXMODE_8_5_SA TA_G1G2	<p>R_RXMODE[8:5] controls receiver equalizer peaking frequency. See Table 12 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R_RXMODE_8_5</th> </tr> </thead> <tbody> <tr><td>0</td><td>1010</td></tr> <tr><td>1</td><td>0111</td></tr> <tr><td>2</td><td>0110</td></tr> <tr><td>3</td><td>0101</td></tr> <tr><td>4</td><td>0100</td></tr> <tr><td>5</td><td>0011</td></tr> <tr><td>6</td><td>0010</td></tr> <tr><td>7</td><td>0001</td></tr> <tr><td>8</td><td>0000</td></tr> <tr><td>9</td><td>0000</td></tr> <tr><td>10</td><td>0000</td></tr> <tr><td>11</td><td>0000</td></tr> <tr><td>12</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_8_5	0	1010	1	0111	2	0110	3	0101	4	0100	5	0011	6	0010	7	0001	8	0000	9	0000	10	0000	11	0000	12	0000	1010
Level	R_RXMODE_8_5																															
0	1010																															
1	0111																															
2	0110																															
3	0101																															
4	0100																															
5	0011																															
6	0010																															
7	0001																															
8	0000																															
9	0000																															
10	0000																															
11	0000																															
12	0000																															
16	R/W	RESERVED		1																												
[15:13]	R/W	RESERVED		001																												
12	R/W	RESERVED		1																												
11	R/W	RESERVED		1																												
[10:9]	R/W	R_RXMODE_14_13_ SAS_G1G2	<p>The RXMODE_14/13 Bit controls the receiver linear equalization in conjunction with the RXMODE_12_5 bits.</p> <p>Refer to the Peaking Mode Gain tables in the SPC 8x6G Hardware Specification for valid settings.</p>	11																												

Bit(s)	Access	Name	Description	Reset State																												
[8:5]	R/W	R_RXMODE_12_9_SAS_G1G2	<p>R_RXMODE[12:9] controls receiver equalizer peaking frequency. See Table 12 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R_RXMODE_12_9</th> </tr> </thead> <tbody> <tr><td>0</td><td>1011</td></tr> <tr><td>1</td><td>1001</td></tr> <tr><td>2</td><td>1001</td></tr> <tr><td>3</td><td>1001</td></tr> <tr><td>4</td><td>1001</td></tr> <tr><td>5</td><td>1001</td></tr> <tr><td>6</td><td>1001</td></tr> <tr><td>7</td><td>1001</td></tr> <tr><td>8</td><td>1001</td></tr> <tr><td>9</td><td>0011</td></tr> <tr><td>10</td><td>0010</td></tr> <tr><td>11</td><td>0001</td></tr> <tr><td>12</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_12_9	0	1011	1	1001	2	1001	3	1001	4	1001	5	1001	6	1001	7	1001	8	1001	9	0011	10	0010	11	0001	12	0000	1011
Level	R_RXMODE_12_9																															
0	1011																															
1	1001																															
2	1001																															
3	1001																															
4	1001																															
5	1001																															
6	1001																															
7	1001																															
8	1001																															
9	0011																															
10	0010																															
11	0001																															
12	0000																															
[4:1]	R/W	R_RXMODE_8_5_SAS_G1G2	<p>R_RXMODE[8:5] controls receiver equalizer peaking frequency. See Table 12 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R_RXMODE_8_5</th> </tr> </thead> <tbody> <tr><td>0</td><td>1010</td></tr> <tr><td>1</td><td>0111</td></tr> <tr><td>2</td><td>0110</td></tr> <tr><td>3</td><td>0101</td></tr> <tr><td>4</td><td>0100</td></tr> <tr><td>5</td><td>0011</td></tr> <tr><td>6</td><td>0010</td></tr> <tr><td>7</td><td>0001</td></tr> <tr><td>8</td><td>0000</td></tr> <tr><td>9</td><td>0000</td></tr> <tr><td>10</td><td>0000</td></tr> <tr><td>11</td><td>0000</td></tr> <tr><td>12</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_8_5	0	1010	1	0111	2	0110	3	0101	4	0100	5	0011	6	0010	7	0001	8	0000	9	0000	10	0000	11	0000	12	0000	1010
Level	R_RXMODE_8_5																															
0	1010																															
1	0111																															
2	0110																															
3	0101																															
4	0100																															
5	0011																															
6	0010																															
7	0001																															
8	0000																															
9	0000																															
10	0000																															
11	0000																															
12	0000																															
0	R/W	RESERVED		1																												

10.5.8 Receiver Per Port Configuration 1 SAS_SATA G3 Register

Mnemonic: RX_PPC_SAS_SATA_G3

Address: (via MEMBASE-III):

Using shifted destination address 0x2_0000: 0x114 + 0x100*N(N=0:7)

Description:

Table 275 Receiver Per Port Configuration 1 SAS_SATA G3 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:29]	R/W	RESERVED	TBD	001
28	R/W	R_RXMODE_4_SAS_2	<p>R_RXMODE[4] enables peaking in the RXDP/RXDN inputs for SAS 6 Gbps operation. When set to logic:</p> <p>1: The channel equalization (peaking) circuit is enabled.</p> <p>0: The channel equalization (peaking) circuit is disabled, and the R_RXMODE[12:5] Registers are ignored.</p> <p>R_RXMODE[4] enables fixed equalization.</p>	0
27	R/W	TRS_MODE_0_SATA_G3	<p>Mode bus to control slice functionality. TRS_MODE[0] selects which of the DFE or DCRU SIPO is driving the output interface. When set to logic:</p> <p>1: Data comes from the DCRU. This turns off the DFE and its clock reference in the DCRU.</p> <p>0: Data comes from the DFE (default). This turns off the DCRU SIPO.</p>	1
[26:25]	R/W	R_RXMODE_14_13_SATA_G3	<p>The RXMODE_14/13 Bit controls the receiver linear equalization in conjunction with the RXMODE_12_5 bits.</p> <p>Refer to the Peaking Mode Gain tables in the SPC 8x6G Hardware Specification for valid settings.</p>	11

Bit(s)	Access	Name	Description	Reset State																										
[24:21]	R/W	R_RXMODE_12_9_SATA_G3	<p>R_RXMODE[12:9] controls receiver equalizer peaking frequency. See Table 13 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th><th>R_RXMODE_12_9</th></tr> </thead> <tbody> <tr><td>0</td><td>1010</td></tr> <tr><td>1</td><td>0101</td></tr> <tr><td>2</td><td>0101</td></tr> <tr><td>3</td><td>0101</td></tr> <tr><td>4</td><td>0101</td></tr> <tr><td>5</td><td>0101</td></tr> <tr><td>6</td><td>0101</td></tr> <tr><td>7</td><td>0100</td></tr> <tr><td>8</td><td>0011</td></tr> <tr><td>9</td><td>0010</td></tr> <tr><td>10</td><td>0001</td></tr> <tr><td>11</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_12_9	0	1010	1	0101	2	0101	3	0101	4	0101	5	0101	6	0101	7	0100	8	0011	9	0010	10	0001	11	0000	1010
Level	R_RXMODE_12_9																													
0	1010																													
1	0101																													
2	0101																													
3	0101																													
4	0101																													
5	0101																													
6	0101																													
7	0100																													
8	0011																													
9	0010																													
10	0001																													
11	0000																													
[20:17]	R/W	R_RXMODE_8_5_SATA_G3	<p>R_RXMODE[8:5] controls receiver equalizer peaking frequency. See Table 13 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th><th>R_RXMODE_8_5</th></tr> </thead> <tbody> <tr><td>0</td><td>1010</td></tr> <tr><td>1</td><td>0101</td></tr> <tr><td>2</td><td>0100</td></tr> <tr><td>3</td><td>0011</td></tr> <tr><td>4</td><td>0010</td></tr> <tr><td>5</td><td>0001</td></tr> <tr><td>6</td><td>0000</td></tr> <tr><td>7</td><td>0000</td></tr> <tr><td>8</td><td>0000</td></tr> <tr><td>9</td><td>0000</td></tr> <tr><td>10</td><td>0000</td></tr> <tr><td>11</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_8_5	0	1010	1	0101	2	0100	3	0011	4	0010	5	0001	6	0000	7	0000	8	0000	9	0000	10	0000	11	0000	1010
Level	R_RXMODE_8_5																													
0	1010																													
1	0101																													
2	0100																													
3	0011																													
4	0010																													
5	0001																													
6	0000																													
7	0000																													
8	0000																													
9	0000																													
10	0000																													
11	0000																													
16	R/W	RESERVED		1																										
[15:13]	R/W	DFE_MODE_2_0_SAS_G3	Controls number of DFE taps. Default DFE tap count is 5.	101																										

Bit(s)	Access	Name	Description	Reset State																										
12	R/W	TRS_MODE_0_SAS2	<p>Mode bus to control slice functionality for SAS 6 Gbps operation. TRS_MODE[0] selects which of the DFE or DCRU SIPO is driving the output interface. When set to logic:</p> <p>1: Data comes from the DCRU. This turns off the DFE and its clock reference in the DCRU.</p> <p>0: Data comes from the DFE (default). This turns off the DCRU SIPO.</p>	0																										
11	R/W	TRS_MODE_0_SAS_G3	TBD	1																										
10	R/W	R_RXMODE_14_13_SAS_G3	<p>The RXMODE_14/13 Bit controls the receiver linear equalization in conjunction with the RXMODE_12_5 bits.</p> <p>Refer to the Peaking Mode Gain tables in the SPC 8x6G Hardware Specification for valid settings.</p>	11																										
[8:5]	R/W	R_RXMODE_12_9_SAS_G3	<p>R_RXMODE[12:9] controls receiver equalizer peaking frequency. See Table 13 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R_RXMODE_12_9</th> </tr> </thead> <tbody> <tr><td>0</td><td>1010</td></tr> <tr><td>1</td><td>0101</td></tr> <tr><td>2</td><td>0101</td></tr> <tr><td>3</td><td>0101</td></tr> <tr><td>4</td><td>0101</td></tr> <tr><td>5</td><td>0101</td></tr> <tr><td>6</td><td>0101</td></tr> <tr><td>7</td><td>0100</td></tr> <tr><td>8</td><td>0011</td></tr> <tr><td>9</td><td>0010</td></tr> <tr><td>10</td><td>0001</td></tr> <tr><td>11</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_12_9	0	1010	1	0101	2	0101	3	0101	4	0101	5	0101	6	0101	7	0100	8	0011	9	0010	10	0001	11	0000	1010
Level	R_RXMODE_12_9																													
0	1010																													
1	0101																													
2	0101																													
3	0101																													
4	0101																													
5	0101																													
6	0101																													
7	0100																													
8	0011																													
9	0010																													
10	0001																													
11	0000																													

Bit(s)	Access	Name	Description	Reset State																										
[4:1]	R/W	R_RXMODE_8_5_SA_S_G3	<p>R_RXMODE[8:5] controls receiver equalizer peaking frequency. See Table 13 in the SPC 8x6G Hardware Specification (PMC-2080174) for peaking levels.</p> <table> <thead> <tr> <th>Level</th> <th>R_RXMODE_8_5</th> </tr> </thead> <tbody> <tr><td>0</td><td>1010</td></tr> <tr><td>1</td><td>0101</td></tr> <tr><td>2</td><td>0100</td></tr> <tr><td>3</td><td>0011</td></tr> <tr><td>4</td><td>0010</td></tr> <tr><td>5</td><td>0001</td></tr> <tr><td>6</td><td>0000</td></tr> <tr><td>7</td><td>0000</td></tr> <tr><td>8</td><td>0000</td></tr> <tr><td>9</td><td>0000</td></tr> <tr><td>10</td><td>0000</td></tr> <tr><td>11</td><td>0000</td></tr> </tbody> </table>	Level	R_RXMODE_8_5	0	1010	1	0101	2	0100	3	0011	4	0010	5	0001	6	0000	7	0000	8	0000	9	0000	10	0000	11	0000	1010
Level	R_RXMODE_8_5																													
0	1010																													
1	0101																													
2	0100																													
3	0011																													
4	0010																													
5	0001																													
6	0000																													
7	0000																													
8	0000																													
9	0000																													
10	0000																													
11	0000																													
0	R/W	RESERVED		1																										

10.5.9 Global Configuration 1 Register

Mnemonic: TRS_GLOBAL_CFG

Address: (via MEMBASE-III):
Using shifted destination address $0x2_0000: 0x120 + 0x100*N(N=0:7)$

Description:

Table 276 Global Configuration 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	R/W	RESERVED		00
[29:28]	R/W	RESERVED		00
[27:26]	R/W	RESERVED		00
[25:24]	R/W	RESERVED		00
[23:22]	R/W	RESERVED		00
[21:20]	R/W	RESERVED		00
[19:18]	—	UNUSED		XX
17	R/W	RESERVED		0

Bit(s)	Access	Name	Description	Reset State
16	R/W	RESERVED		0
15	R/W	RESERVED		0
[14:4]	R/W	RESERVED		000 0000 0000
3	R/W	RESERVED		0
[2:0]	R/W	TRS_LBSEL	Loop back selection signal. TRS_LBSEL[2:0] Operation Mode 000 Normal reception and transmission 001 Metallic loop back from DFE input to transmitter input. 010 Metallic loop back from DCRU input to transmitter input.	000

10.5.10 Transmitter Configuration 1 Register

Mnemonic: TX_CFG_1

Address: (via MEMBASE-III):
Using shifted destination address $0x2_0000: 0x128 + 0x100*N(N=0:7)$

Description:

Table 277 Transmitter Configuration 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:28]	R/W	RESERVED		0000
27	—	UNUSED		X

Bit(s)	Access	Name	Description	Reset State																											
26	R/W	T_MODE_15_SATA_G3	<p>For SATA 6 Gbps operation.</p> <p>Used in conjunction with T_CNTL[7:6] bits in the OSSP – Transmitter Per Port Configuration 1 SAS_SATA G3 Register to set the pre-driver current.</p> <table> <thead> <tr> <th>T_MODE</th> <th>T_CTRL[7:6]</th> <th>Current (µA)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>00</td> <td>100</td> </tr> <tr> <td>0</td> <td>01</td> <td>80</td> </tr> <tr> <td>0</td> <td>10</td> <td>70</td> </tr> <tr> <td>0</td> <td>11</td> <td>60</td> </tr> <tr> <td>1</td> <td>00</td> <td>90</td> </tr> <tr> <td>1</td> <td>01</td> <td>110</td> </tr> <tr> <td>1</td> <td>10</td> <td>120</td> </tr> <tr> <td>1</td> <td>11</td> <td>130</td> </tr> </tbody> </table> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	T_MODE	T_CTRL[7:6]	Current (µA)	0	00	100	0	01	80	0	10	70	0	11	60	1	00	90	1	01	110	1	10	120	1	11	130	0
T_MODE	T_CTRL[7:6]	Current (µA)																													
0	00	100																													
0	01	80																													
0	10	70																													
0	11	60																													
1	00	90																													
1	01	110																													
1	10	120																													
1	11	130																													
25	R/W	T_MODE_15_SATA_G2	<p>For SATA 3 Gbps operation.</p> <p>Used in conjunction with T_CNTL[7:6] bits in the OSSP – Transmitter Per Port Configuration 1 SAS_SATA G2 Register to set the pre-driver current.</p> <table> <thead> <tr> <th>T_MODE</th> <th>T_CTRL[7:6]</th> <th>Current (µA)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>00</td> <td>100</td> </tr> <tr> <td>0</td> <td>01</td> <td>80</td> </tr> <tr> <td>0</td> <td>10</td> <td>70</td> </tr> <tr> <td>0</td> <td>11</td> <td>60</td> </tr> <tr> <td>1</td> <td>00</td> <td>90</td> </tr> <tr> <td>1</td> <td>01</td> <td>110</td> </tr> <tr> <td>1</td> <td>10</td> <td>120</td> </tr> <tr> <td>1</td> <td>11</td> <td>130</td> </tr> </tbody> </table> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	T_MODE	T_CTRL[7:6]	Current (µA)	0	00	100	0	01	80	0	10	70	0	11	60	1	00	90	1	01	110	1	10	120	1	11	130	0
T_MODE	T_CTRL[7:6]	Current (µA)																													
0	00	100																													
0	01	80																													
0	10	70																													
0	11	60																													
1	00	90																													
1	01	110																													
1	10	120																													
1	11	130																													

Bit(s)	Access	Name	Description	Reset State																											
24	R/W	T_MODE_15_SATA_G1	<p>For SATA 1.5 Gbps operation.</p> <p>Used in conjunction with T_CNTL[7:6] bits in the OSSP – Transmitter Per Port Configuration 1 SAS_SATA G1 Register to set the pre-driver current.</p> <table> <thead> <tr> <th>T_MODE</th> <th>T_CTRL[7:6]</th> <th>Current (µA)</th> </tr> </thead> <tbody> <tr><td>0</td><td>00</td><td>100</td></tr> <tr><td>0</td><td>01</td><td>80</td></tr> <tr><td>0</td><td>10</td><td>70</td></tr> <tr><td>0</td><td>11</td><td>60</td></tr> <tr><td>1</td><td>00</td><td>90</td></tr> <tr><td>1</td><td>01</td><td>110</td></tr> <tr><td>1</td><td>10</td><td>120</td></tr> <tr><td>1</td><td>11</td><td>130</td></tr> </tbody> </table> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	T_MODE	T_CTRL[7:6]	Current (µA)	0	00	100	0	01	80	0	10	70	0	11	60	1	00	90	1	01	110	1	10	120	1	11	130	0
T_MODE	T_CTRL[7:6]	Current (µA)																													
0	00	100																													
0	01	80																													
0	10	70																													
0	11	60																													
1	00	90																													
1	01	110																													
1	10	120																													
1	11	130																													
23	—	UNUSED		X																											
22	R/W	T_MODE_15_SAS_G3	<p>For SAS 6 Gbps operation.</p> <p>Used in conjunction with T_CNTL[7:6] bits in the OSSP – Transmitter Per Port Configuration 1 SAS_SATA G3 Register to set the pre-driver current.</p> <table> <thead> <tr> <th>T_MODE</th> <th>T_CTRL[7:6]</th> <th>Current (µA)</th> </tr> </thead> <tbody> <tr><td>0</td><td>00</td><td>100</td></tr> <tr><td>0</td><td>01</td><td>80</td></tr> <tr><td>0</td><td>10</td><td>70</td></tr> <tr><td>0</td><td>11</td><td>60</td></tr> <tr><td>1</td><td>00</td><td>90</td></tr> <tr><td>1</td><td>01</td><td>110</td></tr> <tr><td>1</td><td>10</td><td>120</td></tr> <tr><td>1</td><td>11</td><td>130</td></tr> </tbody> </table> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	T_MODE	T_CTRL[7:6]	Current (µA)	0	00	100	0	01	80	0	10	70	0	11	60	1	00	90	1	01	110	1	10	120	1	11	130	0
T_MODE	T_CTRL[7:6]	Current (µA)																													
0	00	100																													
0	01	80																													
0	10	70																													
0	11	60																													
1	00	90																													
1	01	110																													
1	10	120																													
1	11	130																													

Bit(s)	Access	Name	Description	Reset State																											
21	R/W	T_MODE_15_SAS_G2	<p>For SAS 3 Gbps operation.</p> <p>Used in conjunction with T_CNTL[7:6] bits in the OSSP – Transmitter Per Port Configuration 1 SAS_SATA G2 Register to set the pre-driver current.</p> <table> <thead> <tr> <th>T_MODE</th> <th>T_CTRL[7:6]</th> <th>Current (µA)</th> </tr> </thead> <tbody> <tr><td>0</td><td>00</td><td>100</td></tr> <tr><td>0</td><td>01</td><td>80</td></tr> <tr><td>0</td><td>10</td><td>70</td></tr> <tr><td>0</td><td>11</td><td>60</td></tr> <tr><td>1</td><td>00</td><td>90</td></tr> <tr><td>1</td><td>01</td><td>110</td></tr> <tr><td>1</td><td>10</td><td>120</td></tr> <tr><td>1</td><td>11</td><td>130</td></tr> </tbody> </table> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	T_MODE	T_CTRL[7:6]	Current (µA)	0	00	100	0	01	80	0	10	70	0	11	60	1	00	90	1	01	110	1	10	120	1	11	130	0
T_MODE	T_CTRL[7:6]	Current (µA)																													
0	00	100																													
0	01	80																													
0	10	70																													
0	11	60																													
1	00	90																													
1	01	110																													
1	10	120																													
1	11	130																													
20	R/W	T_MODE_15_SAS_G1	<p>For SAS 1.5 Gbps operation.</p> <p>Used in conjunction with T_CNTL[7:6] bits in the OSSP – Transmitter Per Port Configuration 1 SAS_SATA G1 Register to set the pre-driver current.</p> <table> <thead> <tr> <th>T_MODE</th> <th>T_CTRL[7:6]</th> <th>Current (µA)</th> </tr> </thead> <tbody> <tr><td>0</td><td>00</td><td>100</td></tr> <tr><td>0</td><td>01</td><td>80</td></tr> <tr><td>0</td><td>10</td><td>70</td></tr> <tr><td>0</td><td>11</td><td>60</td></tr> <tr><td>1</td><td>00</td><td>90</td></tr> <tr><td>1</td><td>01</td><td>110</td></tr> <tr><td>1</td><td>10</td><td>120</td></tr> <tr><td>1</td><td>11</td><td>130</td></tr> </tbody> </table> <p>These bits can be used to optimize the transmitter for EMI. Higher pre-driver current will result in larger transmit amplitude at the cost of greater EMI emissions.</p>	T_MODE	T_CTRL[7:6]	Current (µA)	0	00	100	0	01	80	0	10	70	0	11	60	1	00	90	1	01	110	1	10	120	1	11	130	0
T_MODE	T_CTRL[7:6]	Current (µA)																													
0	00	100																													
0	01	80																													
0	10	70																													
0	11	60																													
1	00	90																													
1	01	110																													
1	10	120																													
1	11	130																													
19	—	UNUSED		X																											
18	R/W	RESERVED		0																											

Bit(s)	Access	Name	Description	Reset State
[17:16]	R/W	RESERVED		00
[15:14]	—	UNUSED		XX
[13:12]	R/W	T_MODE_1_0_SATA_G3	For SATA 6 Gbps operation. Controls the edge rate for the transmit signal. T_MODE_1_0 Edge Rate (ps) 00 57 01 68 10 82 11 97	00
[11:10]	R/W	T_MODE_1_0_SATA_G2	For SATA 3 Gbps operation. Controls the edge rate for the transmit signal. T_MODE_1_0 Edge Rate (ps) 00 57 01 68 10 82 11 97	10
[9:8]	R/W	T_MODE_1_0_SATA_G1	For SATA 1.5 Gbps operation. Controls the edge rate for the transmit signal. T_MODE_1_0 Edge Rate (ps) 00 57 01 68 10 82 11 97	11
[7:6]	—	UNUSED		XX
[5:4]	R/W	T_MODE_1_0_SAS_G3	For SAS 6 Gbps operation. Controls the edge rate for the transmit signal. T_MODE_1_0 Edge Rate (ps) 00 57 01 68 10 82 11 97	00

Bit(s)	Access	Name	Description	Reset State
[3:2]	R/W	T_MODE_1_0_SAS_G2	For SAS 3 Gbps operation. Controls the edge rate for the transmit signal. T_MODE_1_0 Edge Rate (ps) 00 57 01 68 10 82 11 97	10
[1:0]	R/W	T_MODE_1_0_SAS_G1	For SAS 1.5 Gbps operation. Controls the edge rate for the transmit signal. T_MODE_1_0 Edge Rate (ps) 00 57 01 68 10 82 11 97	11

10.5.11 Receiver Configuration 1 Register

Mnemonic: RX_CFG_1

Address: (via MEMBASE-III):
Using shifted destination address $0x2_0000: 0x130 + 0x100*N(N=0:7)$

Description:

Table 278 Receiver Configuration 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:25]	—	UNUSED		XXX XXXX
24	R/W	RESERVED		0
[23:21]	—	UNUSED		XXX
[20:15]	R/W	RESERVED		00 0000
[14:4]	—	UNUSED		XXX XXXX XXXX

Bit(s)	Access	Name	Description	Reset State
3:0	R/W	R_RXMODE_3_0	<p>R_RXMODE[3] is reserved and must be left at its default value for normal operation.</p> <p>R_RXMODE[2:0] control mode inputs for the receiver.</p> <p>R_RXMODE[2:0]: Input bias control</p> <ul style="list-style-type: none"> 000: 100 Ohm differential termination to AVD12; on-chip AC coupling is active. 001: 50 Ohm single-ended termination to AVD12; on-chip AC coupling is active. 010: 50 Ohm single-ended termination to 250 mV; on-chip AC coupling is active. 011: 50 Ohm single-ended termination to AVS; on-chip AC coupling is active. (default) 100: 50 Ohm single-ended termination to AVD12/2; on-chip AC coupling is active. 101: On-chip AC coupling is bypassed; termination is connected to internal common-mode. <p>For reliability reasons, this mode is illegal when the input voltage level is higher than (TBD)V.</p> <p>111 High Impedance. These bits must be independently controlled during JTAG testing.</p>	0100

10.5.12 Receiver Configuration 2 Register

Mnemonic: RX_CFG_2

Address: (via MEMBASE-III):
Using shifted destination address 0x2_0000: 0x134 + 0x100*N(N=0:7)

Description: Reserved

Table 279 Receiver Configuration 2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:26]	—	UNUSED		XX XXXX
25	R/W	R_ATMSB	Reserved	1
24	R/W	R_ENB	Reserved	0
[23:16]	—	UNUSED		XXXX XXXX

Bit(s)	Access	Name	Description	Reset State
[15:8]	R/W	R_LOSMODE_11_4	Reserved	0000 0010
[7:4]	R/W	R_LOSMODE_3_0_SATA	Reserved	0010
[3:0]	R/W	R_LOSMODE_3_0_SAS	Reserved	0100

10.6 SAS/SATA PHY Layer Registers

The 32-bit BAR specified at PCI configuration address 0x20 (MEMBASE-III) contains the following chip registers used for configuring the SPC 8x6G SAS/SATA PHY layer. The following subsections provide the descriptions for these registers.

The host must perform MEMBASE-III inbound window-shifting aligned on 64 KB as shown in Section 2.6.1, “[MEMBASE-III Inbound Window Shifting](#)”. The remaining space provided in the offset must be used as the offset for reading/writing the register using MEMBASE-III.

Table 280 SAS/SATA PHY Layer Address Map

Offset	MEMBASE	Name
0x3_0000: 0x1010 + 0x4000*N (N=0:3) 0x4_0000: 0x1010 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Connection Status Register
0x3_0000: 0x1014 + 0x4000*N (N=0:3) 0x4_0000: 0x1014 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Error Interval Thresholds Register
0x3_0000: 0x1028 + 0x4000*N (N=0:3) 0x4_0000: 0x1028 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Interrupt Status 1 Register
0x3_0000: 0x102C + 0x4000*N (N=0:3) 0x4_0000: 0x102C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Interrupt Values 1 Register
0x3_0000: 0x1030 + 0x4000*N (N=0:3) 0x4_0000: 0x1030 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Counter Configuration Register
0x3_0000: 0x1034 + 0x4000*N (N=0:3) 0x4_0000: 0x1034 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Invalid DWord Count Register
0x3_0000: 0x1038 + 0x4000*N (N=0:3) 0x4_0000: 0x1038 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Disparity Error Count Register
0x3_0000: 0x103C + 0x4000*N (N=0:3) 0x4_0000: 0x103C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Code Violation Error Count Register
0x3_0000: 0x1040 + 0x4000*N (N=0:3) 0x4_0000: 0x1040 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Loss of DWord Synchronization Count Register
0x3_0000: 0x1044 + 0x4000*N (N=0:3) 0x4_0000: 0x1044 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	PHY Reset Failed Count Register

Offset	MEMBASE	Name
0x3_0000: 0x1048 + 0x4000*N (N=0:3) 0x4_0000: 0x1048 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	PRBS and Test Pattern Error Count Register
0x3_0000: 0x104C + 0x4000*N (N=0:3) 0x4_0000: 0x104C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	SAS2 RX Error Count Register
0x3_0000: 0x1050 + 0x4000*N (N=0:3) 0x4_0000: 0x1050 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Diagnostics Configuration Register
0x3_0000: 0x1054 + 0x4000*N (N=0:3) 0x4_0000: 0x1054 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Error Insertion Register 1
0x3_0000: 0x1058 + 0x4000*N (N=0:3) 0x4_0000: 0x1058 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Error Insertion Register 2
0x3_0000: 0x1060 + 0x4000*N (N=0:3) 0x4_0000: 0x1060 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Test Pattern Insertion Word 1_1 Register
0x3_0000: 0x1064 + 0x4000*N (N=0:3) 0x4_0000: 0x1064 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Test Pattern Insertion Word 1_2 Register
0x3_0000: 0x1068 + 0x4000*N (N=0:3) 0x4_0000: 0x1068 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Test Pattern Insertion Word 2_1 Register
0x3_0000: 0x106C + 0x4000*N (N=0:3) 0x4_0000: 0x106C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Test Pattern Insertion Word 2_2 Register
0x3_0000: 0x1074 + 0x4000*N (N=0:3) 0x4_0000: 0x1074 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	SAS 2 Settings (Local) Register
0x3_0000: 0x20A0 + 0x4000*N(N=0:3) 0x4_0000: 0x20A0 + 0x4000*N(N=4:7)	MEMBASE-III	Maximum AIP Allowed
0x3_0000: 0x30B4 + 0x4000*N(N=0:3) 0x4_0000: 0x30B4 + 0x4000*N(N=4:7)	MEMBASE-III	Open Retry Interval

10.6.1 Connection Status Register

Mnemonic: REG4

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1010 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1010 + 0x4000*(N-4) (N=4:7)

Description:

Table 281 Connection Status Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:20]	—	UNUSED		XXXX XXXX XXXX
19	R	SAS2_MUX_EN	<p>SAS2 Muxing Enable indication. When read as logic: 1: The OOB sequence has determined that muxing should be enabled. 0: The OOB sequence has determined that muxing should not be enabled.</p> <p>The SAS2_MUX_EN output pin duplicates the functionality of this register bit.</p>	X
18	R	SAS2_SSC_TYP	<p>SAS2 Spread Spectrum Clocking Type indication. When read as logic: 1: The OOB sequence has negotiated center-spread spectrum clocking with an attached SAS-2 device. 0: The OOB sequence has negotiated down-spread spectrum clocking, or has not negotiated spread spectrum clocking, with an attached SAS-2 device.</p> <p>The SAS2_SSC_TYP output pin duplicates the functionality of this register bit.</p>	X
17	R	SAS2_SSC_EN	<p>SAS2 Spread Spectrum Clocking indication. When read as logic: 1: The OOB sequence has negotiated spread spectrum clocking with an attached SAS-2 device. 0: The OOB sequence has not negotiated spread spectrum clocking with an attached SAS-2 device.</p> <p>The SAS2_SSC_EN output pin duplicates the functionality of this register bit.</p>	X

Bit(s)	Access	Name	Description	Reset State
16	R	SAS2_EN	SAS2 device attached indication. When read as logic: 1: The OOB sequence has detected an attached SAS-2 device. 0: The OOB sequence has not detected an attached SAS-2 device. The SAS2_EN output pin duplicates the functionality of this register bit.	X
[15:8]	—	UNUSED		XXXX XXXX
7	R	SAS_ATTACHED	SAS device attached indication. When read as logic: 1: The OOB sequence has detected an attached SAS device. 0: The OOB sequence has not detected an attached SAS device. The SAS_ATTACHED output pin duplicates the functionality of this register bit.	X
6	R	SATA_ATTACHED	SATA device attached indication. When read as logic: 1: The OOB sequence has detected an attached SATA device. 0: The OOB sequence has not detected an attached SATA device. The SATA_ATTACHED output pin duplicates the functionality of this register bit.	X
5	R	DEV_PRESENT	Device Present indication. When read as logic: 1: Indicates that a device is connected, as a COMINIT/COMRESET or COMSAS OOB signal has been detected. 0: Indicates that no device is connected, as no COMINIT/COMRESET or COMSAS OOB signal has been detected. The DEV_PRESENT output pin duplicates the functionality of this register bit.	X
4	R	PHY_RST_LIM_SAT	PHY Reset Limit Saturation indication. When read as logic: 1: Indicates that the PHY has reached the PHY reset maximum limit as specified by PHY_RST_LIM[6:0]. 0: Indicates that the PHY has not reached the PHY reset maximum limit as specified by PHY_RST_LIM[6:0].	X
3	—	UNUSED		X

Bit(s)	Access	Name	Description	Reset State
[2:0]	R	PHY_RATE	<p>PHY_RATE[2:0] indicates the final negotiated line rate resulting from the PHY reset sequence. It uses one-hot encoding and is valid only when PHYRDY = '1'.</p> <p>When PHYRDY = '0', PHY_RATE[2:0] is set to '000'. This register reflects the PHY_RATE[2:0] output pin value.</p> <p>When read as logic:</p> <ul style="list-style-type: none"> 001: The PHY has negotiated a line rate of 1.5 Gbps. 010: The PHY has negotiated a line rate of 3.0 Gbps. 100: The PHY has negotiated a line rate of 6.0 Gbps. All other PHY_RATE[2:0] values are invalid and indicate that the PHY has not reached PHYRDY. <p>The PHY_RATE[2:0] output pins duplicate the functionality of this register bit.</p>	XXX

10.6.2 Error Interval Thresholds Register

Mnemonic: REG5

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1014 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1014 + 0x4000*(N-4) (N=4:7)

Description:

Table 282 Error Interval Thresholds Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	—	UNUSED		XXXX XXXX XXXX XXXX
[15:8]	R/W	CODE_VIOL_INTRVL_THR	The code violation interval threshold (CODE_VIOL_INTRVL_THR[7:0]) specifies the number of allowable code violation errors that can be tolerated.	0000 0000
[7:0]	R/W	DISP_INTRVL_THR	The disparity interval threshold (DISP_INTRVL_THR[7:0]) specifies the number of allowable disparity errors that can be tolerated.	0000 0000

10.6.3 Interrupt Status 1 Register

Mnemonic: INT_STATUS_1

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1028 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1028 + 0x4000*(N-4) (N=4:7)

Description:

Table 283 Interrupt Status 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
31	—	UNUSED		X
30	R	SAS2_RX_SYNC_V	The SAS2_RX_SYNC_V (PRBS Synchronization) bit reflects the current state of SAS2 Training synchronization. When read as logic: 1: Indicates that the SAS2 Training sequence is in sync. 0: Indicates that the SAS2 Training sequence is not in sync.	0
[29:28]	—	UNUSED		XX
27	R	SAS2_RX_TRAINED_V	The SAS2_RX_TRAINED_V (SAS2 Receiver Trained) bit reflects the current state of SAS2 Receiver training. When read as logic: 1: Indicates that the SAS2 receiver has trained. 0: Indicates that the SAS2 receiver has not trained.	0
26	R	SAS2_RX_PARITY_ERR_V	The SAS2_RX_PARITY_ERR_V (SAS2 Receive Parity Error) bit reflects the current state of the SAS2 Parity Error. When read as logic: 1: Indicates that the last settings string received had a parity error. 0: Indicates that the last settings string received did not have a parity error.	0
[25:23]	—	UNUSED		XXX
22	R	TEST_PAT_SYNC_V	The TEST_PAT_SYNC_V (Test Pattern Synchronization) bit reflects the current state of Test Pattern synchronization. When read as logic: 1: Indicates that the TEST_PAT sequence is in sync. 0: Indicates that the TEST_PAT sequence is not in sync.	0

Bit(s)	Access	Name	Description	Reset State
21	—	UNUSED		X
20	R	PRBS_SYNC_V	<p>The PRBS_SYNC_V (PRBS Synchronization) bit reflects the current state of PRBS synchronization. When read as logic:</p> <p>1: Indicates that the PRBS sequence is in sync.</p> <p>0: Indicates that the PRBS sequence is not in sync.</p>	0
19	—	UNUSED		X
18	R	UNPLUG_DET_V	<p>The UNPLUG_DET_V (Unplug Detect) bit reflects the current state of the unplug detection circuitry. When read as logic:</p> <p>1: The UNPLUG_DET_I interrupt is currently asserted.</p> <p>0: The UNPLUG_DET_I interrupt is currently not asserted.</p> <p>Due to the transient nature of the interrupt, it is recommended that the UNPLUG_DET_VAL bit be used instead of this bit.</p>	0
17	R	HOLD_DET_V	<p>The HOLD_DET_V (Hold Detect) bit reflects the current state of the hold detection circuitry. When read as logic:</p> <p>1: The HOLD_DET_I interrupt is currently asserted.</p> <p>0: The HOLD_DET_I interrupt is currently not asserted.</p> <p>Due to the transient nature of the interrupt, it is recommended that the HOLD_DET_VAL bit be used instead of this bit.</p>	0
16	R	RATE_SNOOP_DONE_V	<p>The RATE_SNOOP_DONE_V (Rate Snoop) bit reflects the current state of the rate snooping circuitry. When read as logic:</p> <p>1: The PHY has updated all RATE_DET registers.</p> <p>0: The PHY has not updated all RATE_DET registers, or rate snooping is not enabled.</p>	0

Bit(s)	Access	Name	Description	Reset State
15	R	PHY_RESET_V	The PHY_RESET_V (PHY Reset) bit reflects the current state of the PHY state machine. When read as logic: 1: Indicates that the PHY reset/power-up sequence (i.e., OOB and speed negotiation) is executed due to errors such as loss of DWord alignment or failure to successfully complete speed negotiation. 0: Indicates that there are currently no errors to cause the PHY reset/power-up sequence.	0
14	R	DWS_LOST_V	The DWS_LOST_V (DWord Synchronization Lost) bit reflects the current state of the DWord synchronization logic. When read as logic: 1: Indicates that alignment to DWord boundaries is lost. 0: Indicates that alignment to DWord boundaries is not lost.	0
13	—	UNUSED		X
12	R	PS_PRES_DET_V	The PS_PRES_DET_V (Port Selector Presence Detect) bit reflects the current state of the port selector presence detection logic. When read as logic: 1: Indicates that the SSPL_6G has detected the presence of the port selector 0: Indicates that the SSPL_6G has not detected the presence of the port selector	0
11	—	UNUSED		X
10	R	COMSAS_DET_V	The COMSAS_DET_V (COMSAS Detect) bit reflects the current state of COMSAS detection. When read as logic: 1: Indicates that a COMSAS OOB signal is currently being detected. 0: Indicates that a COMSAS OOB signal is currently not being detected.	0
9	R	COMINIT_DET_V	The COMINIT_DET_V (COMINIT Detect) bit reflects the current state of COMINIT detection. When read as logic: 1: Indicates that a COMINIT OOB signal is currently being detected. 0: Indicates that a COMINIT OOB signal is currently not being detected.	0

Bit(s)	Access	Name	Description	Reset State
8	R	COMWAKE_DET_V	The COMWAKE_DET_V (COMWAKE Detect) bit reflects the current state of COMWAKE detection. When read as logic: 1: Indicates that a COMWAKE OOB signal is currently being detected. 0: Indicates that a COMWAKE OOB signal is currently not being detected.	0
7	R	PHY_RST_FAILED_V	The PHY_RST_FAILED_V (PHY Reset Failure) bit reflects the current state of the PHY Reset Failure/Problem. When read as logic: 1: Indicates that a PHY reset failure has been detected; the OOB sequence has completed but the PHY did not obtain DWord synchronization during the final SAS speed negotiation window. 0: Indicates that the PHY is not encountering a reset failure. This bit is only valid during SAS mode (SAS_SATAB = '1' and PS_MODE = '0').	0
[6:4]	—	UNUSED		XXX
3	R	PHYRDY_V	The PHYRDY_V (PHY Ready Status) bit reflects the current state of the PHY state machine. When read as logic: 1: Indicates that PHY is ready. 0: Indicates that the PHY is not ready.	0
2	R	SPINUP_HOLD_CONF_V	The SPINUP_HOLD_CONF_V (Spin-up Hold Confirmation) bit reflects the current state of the SATA Spin-up/Hold state machine. When read as logic: 1: Indicates that the SSPL_6G has met the conditions of SAS section 6.7.3.9.2, but has not yet achieved SATA attached or SAS attached. 0: Indicates that the SSPL_6G has not met the conditions of SAS section 6.7.3.9.2.	0
[1:0]	—	UNUSED		XX

10.6.4 Interrupt Values 1 Register

Mnemonic: INT_VALUE_1

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x102C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x102C + 0x4000*(N-4) (N=4:7)

Description: This register contains value bits for interrupts that hold state beyond the event itself. This includes PHY_RESET_FAILED, UNPLUG_DET and HOLD_DET. These values should be used instead of the _V bits in register 0x28 to maintain compatibility with previous SAS PHYs.

Table 284 Interrupt Values 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:19]	—	UNUSED		X XXXX XXXX XXXX
18	R	UNPLUG_DET_VAL	The UNPLUG_DET_VAL (Unplug Detect) bit reflects the current state of the unplug detection circuitry. When read as logic: 1: The SSPL has cycled through a hot plug timeout and no response was received; no other devices are attached on the serial link. 0: The register is cleared when a COMINIT, COMSAS, or COMWAKE is received.	0
17	R	HOLD_DET_VAL	The HOLD_DET_VAL (Hold Detect) bit reflects the current state of the hold detection circuitry. When read as logic: 1: The SSPL was expecting a COMWAKE, but the ASYNC_RECov timer has expired or a COMRESET has occurred. The PHY is being kept in SATA HOLD. The register is cleared when a COMWAKE sequence is received. 0: The PHY is not being kept in SATA HOLD. This bit is only valid during SATA mode.	0
[16:8]	—	UNUSED		X XXXX XXXX
7	R	PHY_RST_FAILED_VAL	The PHY_RST_FAILED_V (PHY Reset Failure) bit reflects the current state of the PHY Reset Failure/Problem. When read as logic: 1: Indicates that a PHY reset failure has been detected; the OOB sequence has completed but the PHY did not obtain DWord synchronization during the final	0

Bit(s)	Access	Name	Description	Reset State
			SAS speed negotiation window. Once PHY_RST_FAILED_VAL is set to logic 1 it remains logic 1 until the PHY successfully completes the final speed negotiation window. 0: Indicates that the PHY has not encountered a reset failure, or that the PHY has successfully completed speed negotiation. This bit is only valid during SAS mode (SAS_SATAB = '1' and PS_MODE = '0').	
[6:0]	—	UNUSED		XXX XXXX

10.6.5 Counter Configuration Register

Mnemonic: COUNTER_CONFIGURATION

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1030 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1030 + 0x4000*(N-4) (N=4:7)

Description: The host uses this register to get PHY error counts.

Table 285 Invalid DWord Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:2]	—	UNUSED		XX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
1	R/W	LCLK_CLEAR	LCLK_CLEAR is the count transfer clear enable. When set to logic: 1: A rising edge on either the LCLK register bit or the LCLK input pin will trigger the transfer of all counter values to their holding registers, and subsequently clear the counters. 0: A rising edge on either the LCLK register bit or the LCLK input pin will trigger the transfer of all counter values to their holding registers.	0

Bit(s)	Access	Name	Description	Reset State
0	R/W	LCLK	<p>LCLK is the counter transfer clock.</p> <p>Writing this register bit to logic '1' when LCLK_CLEAR = '0' triggers the transfer of all counter values to their holding registers. This bit is cleared when the transfers are complete.</p> <p>Writing this register bit to logic '1' when LCLK_CLEAR = '1' triggers the transfer of all counter values to their holding registers, and clears all counters. This bit is cleared when the transfers are complete.</p> <p>The LCLK input pin duplicates the functionality of this register bit.</p>	0

10.6.6 Invalid DWord Count Register

Mnemonic: INVAL_DWORD_CNT

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1034 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1034 + 0x4000*(N-4) (N=4:7)

Description:

Table 286 Invalid DWord Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	INVALID_DWORD_CNT	<p>INVALID_DWORD_CNT[31:0] indicates the number of invalid DWords that have been received outside of the PHY reset sequence.</p> <p>INVALID_DWORD_CNT[31:0] saturates at a value of all ones ($2^{32}-1$). This count is only updated after PHYRDY = '1'. When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000 0000 0000 0000 0000

10.6.7 Disparity Error Count Register

Mnemonic: DISP_ERR_CNT

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1038 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1038 + 0x4000*(N-4) (N=4:7)

Description:

Table 287 Disparity Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	DISP_ERR_CNT	DISP_ERR_CNT[31:0] indicates the number of DWords containing running disparity errors that have been received outside of the PHY reset sequence. DISP_ERR_CNT[31:0] saturates at a value of all ones ($2^{32}-1$). This count is only updated after PHYRDY = '1'. When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.	0000 0000 0000 0000 0000 0000 0000 0000

10.6.8 Code Violation Error Count Register

Mnemonic: CODE_VIOL_ERR_CNT

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x103C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x103C + 0x4000*(N-4) (N=4:7)

Description:

Table 288 Code Violation Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	CODE_VIOL_ERR_CNT	<p>CODE_VIOL_ERR_CNT[31:0] indicates the number of DWords containing 8B/10B code violation errors that have been received outside of the PHY reset sequence.</p> <p>CODE_VIOL_ERR_CNT[31:0] saturates at a value of all ones ($2^{32}-1$). This count is only updated after PHYRDY = '1'. When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000 0000 0000 0000 0000

10.6.9 Loss of DWord Synchronization Count Register

Mnemonic: DWS_LOST_CNT

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1040 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1040 + 0x4000*(N-4) (N=4:7)

Description:

Table 289 Loss of DWord Synchronization Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	DWS_LOST_CNT	<p>DWS_LOST_CNT[31:0] indicates the number of times the PHY has lost DWord synchronization and restarted the PHY reset sequence.</p> <p>DWS_LOST_CNT[31:0] saturates at a value of all ones ($2^{32}-1$). This count is only updated after PHYRDY = '1'. When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.</p>	0000 0000 0000 0000 0000 0000 0000 0000

10.6.10 PHY Reset Failed Count Register

Mnemonic: **PHY_RST_FAILED_CNT**

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1044 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1044 + 0x4000*(N-4) (N=4:7)

Description:

Table 290 PHY Reset Failed Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	PHY_RST_FAILED_CNT	PHY_RST_FAILED_CNT[31:0] indicates the number of times the PHY has failed the final SAS speed negotiation window. PHY_RST_FAILED_CNT[31:0] saturates at a value of all ones ($2^{32}-1$). This count is only updated after PHYRDY = '1'. When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.	0000 0000 0000 0000 0000 0000 0000 0000

10.6.11 PRBS and Test Pattern Error Count Register

Mnemonic: PRBS_TEST_PAT_ERR_CNT

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1048 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1048 + 0x4000*(N-4) (N=4:7)

Description:

Table 291 PRBS and Test Pattern Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	R	TEST_PAT_ERR_CNT	TEST_PAT_ERR_CNT[15:0] indicates the number of bit errors that have been received by the Test Pattern checker in the receive data path. TEST_PAT_ERR_CNT[15:0] saturates at a value of all ones ($2^{16}-1$). When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.	0000 0000 0000 0000
[15:0]	R	PRBS_ERR_CNT	PRBS_ERR_CNT[15:0] indicates the number of bit errors that have been received by the PRBS-7 checker in the receive data path. PRBS_ERR_CNT[15:0] saturates at a value of all ones ($2^{16}-1$). When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.	0000 0000 0000 0000

10.6.12 SAS2 RX Error Count Register

Mnemonic: SAS2_RX_ERR_CNT

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x104C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x104C + 0x4000*(N-4) (N=4:7)

Description:

Table 292 SAS2 RX Error Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:16]	—	UNUSED		XXXX XXXX XXXX XXXX
[15:0]	R	SAS2_RX_ERR_CNT	SAS2_RX_ERR_CNT[15:0] indicates the number of errored bytes that have been received by the SAS2 receiver during training. SAS2_RX_ERR_CNT[15:0] saturates at a value of all ones (216-1). When LCLK transitions from low to high, the counts are transferred to their holding registers. If LCLK_CLEAR = '1', the counts are also reset. LCLK should be low before reading this register. This register is invalid if LCLK = '1'.	0000 0000 0000 0000

10.6.13 Diagnostics Configuration Register

Mnemonic: DIAG_CFG

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1050 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1050 + 0x4000*(N-4) (N=4:7)

Description:

Table 293 Diagnostics Configuration Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX
[29:24]	R	ALIGN_POSITION	This register represents the alignment of the receiver's DWord synchronization circuit. This register is intended for diagnostic purposes only.	XX XXXX
[23:21]	—	UNUSED		XXX

Bit(s)	Access	Name	Description	Reset State
20	R/W	RATE_SEL_OVR_EN	This bit is used to override the SSPL_6G rate negotiation. It does not affect the PHY_RATE output (which is the negotiated rate, and is invalid when this bit is set.) When set to logic: 1: Rate select override is enabled. The SSPL_6G's rate select outputs RX_RATE_SEL and TX_RATE_SEL is driven to the values in RATE_SEL_OVR_RX and RATE_SEL_OVR_TX respectively. 0: Rate select override is disabled.	0
[19:18]	R/W	RATE_SEL_OVR_TX	The value of this register is used to drive the transmitter rate output pins and the internal TX clock dividers when RATE_SEL_OVR_EN is logic 1, or during burnin testing (BURNINB = '0').	00
[17:16]	R/W	RATE_SEL_OVR_RX	The value of this register is used to drive the receiver rate output pins and the internal RX clock dividers when RATE_SEL_OVR_EN is logic 1, or during burnin testing (BURNINB = '0').	00
[15:9]	R/W	PRBS_LFSR_LD_VAL	PRBS_LFSR_LD_VAL controls the initial value of the transmit PRBS-7 Linear Feedback Shift Register.	101 0101
8	R/W	PRBS_ERR_INS_EN	This bit is used in debug mode to insert a bit error into the transmit PRBS stream. This bit is self-clearing. The bit error must be set in the Error Insertion Mask Register (ERR_INS_MSK). When set to logic: 1: PRBS-7 bit error insertion is enabled. 0: PRBS-7 bit error insertion is disabled. This bit is only valid when PRBS-7 transmission is enabled (PRBS_TX_EN = '1')	0
7	R/W	PRBS_INVERT	This bit is used in debug mode to invert the transmit PRBS-7 data stream. When set to logic: 1: PRBS-7 is inverted before it is transmitted. 0: PRBS-7 is not inverted before transmission. This bit is only valid when PRBS-7 transmission is enabled (PRBS_TX_EN = '1')	0

Bit(s)	Access	Name	Description	Reset State
6	R/W	PRBS_RX_EN	This bit is used in debug mode to check PRBS-7 in the receive data path. When set to logic: 1: PRBS-7 checking is enabled. 0: PRBS-7 checking is disabled.	0
5	R/W	PRBS_TX_EN	This bit is used in debug mode to insert PRBS-7 into the transmit data path. When set to logic: 1: PRBS-7 insertion is enabled. 0: PRBS-7 insertion is disabled.	0
4	R/W	TEST_PAT_ERR_INS_EN	This bit is used in debug mode to insert a bit error into the transmit Test Pattern stream. This bit is self-clearing. The bit error must be set in the Error Insertion Mask Register (ERR_INS_MSK). When set to logic: 1: Test pattern bit error insertion is enabled. 0: Test pattern bit error insertion is disabled. This bit is only valid when Test Pattern transmission is enabled (TEST_PAT_TX_EN = '1')	0
3	R/W	TEST_PAT_RX_EN	This bit is used in debug mode to check test pattern insertion into the receive data path. When set to logic: 1: Test pattern checking is enabled. 0: Test pattern checking is disabled.	0
2	R/W	TEST_PAT_TX_EN	This bit is used in debug mode to set test pattern insertion into the transmit data path. When set to logic: 1: Test pattern insertion is enabled. 0: Test pattern insertion is disabled.	0
1	R/W	CODE_VIOL_ERR_INS_EN	This bit is used in debug mode to insert a code violation error into the transmit data path. This bit is self-clearing. When set to logic: 1: Code violation error insertion is enabled. 0: Code violation error insertion is disabled.	0
0	R/W	DISP_ERR_INS_EN	This bit is used in debug mode to insert a disparity error into the transmit data path. This bit is self-clearing. When set to logic: 1: Disparity error insertion is enabled. 0: Disparity error insertion is disabled.	0

10.6.14 Error Insertion Register 1

Mnemonic: ERR_INS_1

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1054 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1054 + 0x4000*(N-4) (N=4:7)

Description:

Table 294 Error Insertion Register 1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX
[29:20]	R/W	ERR_INS_MSK_2	ERR_INS_MSK_2 is the 8B/10B-encoded byte 2 mask to indicate which bits in the transmit PRBS-7 or Test Pattern stream should be errored when PRBS_ERR_INS_EN or TEST_PAT_ERR_INS_EN is set to logic 1.	00 0000 0000
[19:10]	R/W	ERR_INS_MSK_1	ERR_INS_MSK_1 is the 8B/10B-encoded byte 1 mask to indicate which bits in the transmit PRBS-7 or Test Pattern stream should be errored when PRBS_ERR_INS_EN or TEST_PAT_ERR_INS_EN is set to logic 1.	00 0000 0000
[9:0]	R/W	ERR_INS_MSK_0	ERR_INS_MSK_0 is the 8B/10B-encoded byte 0 (LSB) mask to indicate which bits in the transmit PRBS-7 or Test Pattern stream should be errored when PRBS_ERR_INS_EN or TEST_PAT_ERR_INS_EN is set to logic 1.	00 0000 0000

10.6.15 Error Insertion Register 2

Mnemonic: ERR_INS_2

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1058 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1058 + 0x4000*(N-4) (N=4:7)

Description:

Table 295 Error Insertion Register 2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX
[29:28]	R/W	CODE_VIOL_ERR_INS_BYT	CODE_VIOL_ERR_INS_BYT is the byte index for the code violation error byte CODE_VIOL_ERR_INS. A value of "00" indicates the LSB, and a value of "11" indicates the MSB.	00
[27:26]	—	UNUSED		XX
[25:16]	R/W	CODE_VIOL_ERR_INS	CODE_VIOL_ERR_INS is the (incorrectly) 8B/10B-encoded byte to be inserted into the transmit data path. When CODE_VIOL_ERR_INS_EN is set to logic 1, CODE_VIOL_ERR_INS is inserted into the transmit data path as configured by this register and the Diagnostics Configuration register	00 0000 0000
[15:10]	—	UNUSED		XX XXXX
[9:0]	R/W	ERR_INS_MSK_3	ERR_INS_MSK_3 is the 8B/10B-encoded byte 3 (MSB) mask to indicate which bits in the transmit PRBS-7 or Test Pattern stream should be errored when PRBS_ERR_INS_EN or TEST_PAT_ERR_INS_EN is set to logic 1.	00 0000 0000

10.6.16 Test Pattern Insertion Word 1_1 Register

Mnemonic: TEST_PAT_INS_1_1

Address: (via MEMBASE-III):
 Using shifted destination address 0x3_0000: 0x1060 + 0x4000*N (N=0:3)
 Using shifted destination address 0x4_0000: 0x1060 + 0x4000*(N-4) (N=4:7)

Description:

Table 296 Test Pattern Insertion Word 1_1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX

Bit(s)	Access	Name	Description	Reset State
[29:20]	R/W	TEST_PAT_INS_0_2	TEST_PAT_INS_0_2 is the 8B/10B-encoded byte 2 of the first test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_0_2 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000
[19:10]	R/W	TEST_PAT_INS_0_1	TEST_PAT_INS_0_1 is the 8B/10B-encoded byte 1 of the first test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_0_1 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000
[9:0]	R/W	TEST_PAT_INS_0_0	TEST_PAT_INS_0_0 is the 8B/10B-encoded byte 0 (LSB) of the first test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_0_0 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.6.17 Test Pattern Insertion Word 1_2 Register

Mnemonic: TEST_PAT_INS_1_2

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1064 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1064 + 0x4000*(N-4) (N=4:7)

Description:

Table 297 Test Pattern Insertion Word 1_2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX

Bit(s)	Access	Name	Description	Reset State
[23:16]	R/W	TEST_PAT_RPT_0	TEST_PAT_RPT_0 is the optional repeat count for the first test pattern DWord. When TEST_PAT_RPT_0 is non-zero, the first test pattern DWord is repeated an additional TEST_PAT_RPT_0 times each test pattern loop, for a total of (TEST_PAT_RPT_0 + 1) insertions of this DWord per loop. When TEST_PAT_RPT_0 is zero, the first test pattern DWord is inserted only once per test pattern loop.	0000 0000
[15:10]	—	UNUSED		XX XXXX
[9:0]	R/W	TEST_PAT_INS_0_3	TEST_PAT_INS_0_3 is the 8B/10B-encoded byte 3 (MSB) of the first test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_0_3 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.6.18 Test Pattern Insertion Word 2_1 Register

Mnemonic: TEST_PAT_INS_2_1

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1068 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1068 + 0x4000*(N-4) (N=4:7)

Description:

Table 298 Test Pattern Insertion Word 2_1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:30]	—	UNUSED		XX
[29:20]	R/W	TEST_PAT_INS_1_2	TEST_PAT_INS_1_2 is the 8B/10B-encoded byte 2 of the second test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_1_2 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

Bit(s)	Access	Name	Description	Reset State
[19:10]	R/W	TEST_PAT_INS_1_1	TEST_PAT_INS_1_1 is the 8B/10B-encoded byte 1 of the second test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_1_1 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000
[9:0]	R/W	TEST_PAT_INS_1_0	TEST_PAT_INS_1_0 is the 8B/10B-encoded byte 0 (LSB) of the second test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_1_0 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.6.19 Test Pattern Insertion Word 2_2 Register

Mnemonic: TEST_PAT_INS_2_2

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x106C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x106C + 0x4000*(N-4) (N=4:7)

Description:

Table 299 Test Pattern Insertion Word 2_2 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:16]	R/W	TEST_PAT_RPT_1	TEST_PAT_RPT_1 is the optional repeat count for the second test pattern DWord. When TEST_PAT_RPT_1 is non-zero, the second test pattern DWord is repeated an additional TEST_PAT_RPT_1 times each test pattern loop, for a total of (TEST_PAT_RPT_1 + 1) insertions of this DWord per loop. When TEST_PAT_RPT_1 is zero, the second test pattern DWord is inserted only once per test pattern loop.	0000 0000
[15:10]	—	UNUSED		XX XXXX

Bit(s)	Access	Name	Description	Reset State
[9:0]	R/W	TEST_PAT_INS_1_3	TEST_PAT_INS_1_3 is the 8B/10B-encoded byte 3 (MSB) of the second test pattern DWord. When TEST_PAT_INS_EN is set to logic 1, TEST_PAT_INS_1_3 is inserted into the transmit data path as configured by this register and the Diagnostics Configuration Register	00 0000 0000

10.6.20 SAS 2 Settings (Local) Register

Mnemonic: SAS2_SETTINGS_LOCAL

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x1074 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x1074 + 0x4000*(N-4) (N=4:7)

Description:

Table 300 SAS 2 Settings (Local) Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	SAS2_SETTINGS_LOCAL	SAS2_SETTINGS is the SAS2 settings pattern. It is fully described in T10 07-091r3.	0000 0000 0000 0000 0011 1111 0000 0011

10.6.21 Maximum AIP Allowed Register

Mnemonic: MAX_AIP

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x20A0 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x20A0 + 0x4000*(N-4) (N=4:7)

Description: After transmitting an OPEN, the SPC 8x6G will break the connection if an excessive number (more than or equal to MAX_AIP) of AIP primitives is received. 0 means there is no limit

Table 301 Maximum AIP Allowed Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	—	MAX_AIP	After transmitting an OPEN, the SPC 8x6G will break the connection if an excessive number (more than or equal to MAX_AIP) of AIP primitives is received. 0 means there is no limit	0000 0000 0000 0000 0000 0000 0000 0000

10.6.22 Open Retry Interval Register

Mnemonic: OPEN_RETRY_INTERVAL_REG

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x30B4 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x30B4 + 0x4000*(N-4) (N=4:7)

Description: This value is the time to wait before retrying sending an open frame again

Table 302 Open Retry Interval Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	OPEN_RETRY_INTE RVAL_REG	Bit 31:8 must be zeros. Bit 7:0: This value is the time to wait in microseconds before retrying to send an open frame. A value of zero means that the SPC 8x6G will retry the open frame as quickly as possible..	xxxx xxxxxxxx xxxxxxxx xxxx 0000 0000

10.7 SAS/SATA Port Adapter Link Registers

The 32-bit BAR specified at PCI configuration address 0x20 (MEMBASE-III) contains the following chip registers used for configuring the SPC 8x6G SAS/SATA port adapter link. The following subsections provide the descriptions for these registers.

The host must perform MEMBASE-III inbound window-shifting aligned on 64 KB as shown in Section 2.6.1, “[MEMBASE-III Inbound Window Shifting](#)”. The remaining space provided in the offset must be used as the offset for reading/writing the register using MEMBASE-III.

Table 303 SAS/SATA Port Adapter Link Address Map

Offset	MEMBASE	Name
0x3_0000: 0x2000 + 0x4000*N (N=0:3) 0x4_0000: 0x2000 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Control Register
0x3_0000: 0x2078 + 0x4000*N (N=0:3) 0x4_0000: 0x2078 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Performance Counter 1 Control Register
0x3_0000: 0x207C + 0x4000*N (N=0:3) 0x4_0000: 0x207C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Performance Counter 1 Threshold Register
0x3_0000: 0x2080 + 0x4000*N (N=0:3) 0x4_0000: 0x2080 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Performance Counter 1 Count Register
0x3_0000: 0x2084 + 0x4000*N (N=0:3) 0x4_0000: 0x2084 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Performance Counter 2 Control Register
0x3_0000: 0x2088 + 0x4000*N (N=0:3) 0x4_0000: 0x2088 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Performance Counter 2 Threshold Register
0x3_0000: 0x208C + 0x4000*N (N=0:3) 0x4_0000: 0x208C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Performance Counter 2 Count Register
0x3_0000: 0x2090 + 0x4000*N (N=0:3) 0x4_0000: 0x2090 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Peak Detector Control Register
0x3_0000: 0x2094 + 0x4000*N (N=0:3) 0x4_0000: 0x2094 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Peak Detector 1 Count Register
0x3_0000: 0x2098 + 0x4000*N (N=0:3) 0x4_0000: 0x2098 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Peak Detector 1 Threshold Register
0x3_0000: 0x209C + 0x4000*N (N=0:3) 0x4_0000: 0x209C + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Timer Enables Register
0x3_0000: 0x20A8 + 0x4000*N (N=0:3) 0x4_0000: 0x20A8 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Rate Control Register
0x3_0000: 0x20BC + 0x4000*N (N=0:3) 0x4_0000: 0x20BC + 0x4000*(N-4) (N=4:7)	MEMBASE-III	Connection Time Register
0x3_0000: 0x20C0 + 0x4000*N (N=0:3) 0x4_0000: 0x20C0 + 0x4000*(N-4) (N=4:7)	MEMBASE-III	General Purpose Register

10.7.1 Control Register

Mnemonic: CONTROL

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x2000 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x2000 + 0x4000*(N-4) (N=4:7)

Description:

Table 304 Control Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	SW_RESET	This bit resets the core logic of the SSPA. When set to logic 1, the SSPA core is held in reset.	0
[30:29]	—	UNUSED		XX
28	R	ID_SUCCESSFUL	When logic 1, indicates link ready for use (i.e., The physical link is ready and the identify frames have been exchanged).	0
27	R/W	TIP_TRIG	When triggered (written to logic 1), causes all read only bits mapped to internal state or counters to update. Self clears when done. Note that the Trig bits can be triggered using a byte write to the most significant byte without having to do any masking (since they cannot be written to logic 0 and there are no RW bits grouped with them.).	0
26	R/W	NOTIFY_TRIG	When triggered (written to logic 1), causes NOTIFY primitives to replace ALIGNs. Notify primitive to use is specified by NOTIFY_TYPE (Register 29H). Self clears when done (i.e., when a NOTIFY primitive has been transmitted). Note that the Trig bits can be triggered using a byte write to the most significant byte without having to do any masking (since they cannot be written to logic 0 and there are no RW bits grouped with them.).	0
25	—	UNUSED		X

Bit(s)	Access	Name	Description	Reset State
24	R/W	TEST_ERROR_TRIG	When triggered (written to logic 1), inserts an error into the transmit path test pattern. Self clears when done. Note that the Trig bits can be triggered using a byte write to the most significant byte without having to do any masking (since they cannot be written to logic 0 and there are no RW bits grouped with them.)	0
23	—	UNUSED		X
[22:21]	R/W	CRC_DATA_CHECK_MODE	Causes the CRC check result on received data frames to be as follows: 0 = normal check 1= always correct 2 = always incorrect, all others reserved When set to 2 or when set to 0 and a bad CRC is detected in a data frame, the Message = 7, MESSAGE_ARG[0] = 1 is received in register AH.	00
[20:19]	R/W	CRC_ADDR_CHECK_MODE	Causes the CRC check result on received address frames to be as follows: 0 = normal check, 1= always correct, 2 = always incorrect, all others reserved. When set to 2 or 0 and a bad CRC is detected in an address frame, the ADDRESS_FRAME_ERR_CRC_I (non-identify frame) or IDENTIFY_FRAME_FAILED_I interrupts are set in register AH.	00
18	R/W	CRC_ADDR_GEN_DISABLE	Setting this bit to 1, generates wrong CRC values in transmitted Address frames for debug purposes.	0
17	R/W	SMP_CRC_ERR	Setting this bit to 1, generates wrong CRC values in transmitted SMP frames for debug purposes.	0
16	R/W	SSP_CRC_ERR	Setting this bit to 1 generates wrong CRC values in transmitted SSP frames for debug purposes.	0
15	R/W	STP_CRC_ERR	Setting this bit to 1 generates wrong CRC values in transmitted STP frames for debug purposes.	0
14	R/W	SSP_DIS_PRIM_IN_FRAM	When logic 1, ACK,NACK,RRDY primitives are not allowed inside frames (incase peer designs can't support this)	0
[13:11]	—	UNUSED		XXX

Bit(s)	Access	Name	Description	Reset State
10	R/W	SEND_AIP_MODE	When logic 1, AIP(NORMAL) is sent every 64 DWORDS after an OPEN is received until and OPEN_ACCEPT or OPEN_REJECT is transmitted.	0
[9:6]	R/W	TEST_MODE	These bits specify which test pattern: 0 = no pattern 1= framed CJPAT 2 = non framed CJPAT 3 to 8 = SATA patterns as defined in the SATA standard Other values are reserved.	0000
5	R/W	FPGA_MODE	When logic 1, indicates all clocks are running at quarter rate for FPGA emulation (so no rate matching align insertion is done, this is the only side effect of slower clocks).	0
4	R/W	SCRAMBLING_DISABLE	When logic 1, turns off datapath scrambling and descrambling for debug purposes.	0
3	R/W	SATA_CONT_DISABLE	Setting this bit to 1, turns off continuous primitive suppression using SATA_CONT for debug purposes	0
2	—	UNUSED		X
1	R/W	LOOPBACK	When logic 1, the output of the RXFIFO is looped back to the input of the TXFIFO. There are no restrictions as to when this mode can be entered or exited.	0
0	R/W	ZONE_FRAME_ENABLE	When logic 1, PMC-Sierra's proprietary broadcast frames (frame type 2) are parsed into the broadcast FIFO. When logic 0, address frames of type 2 are considered as the PHYSICAL address frame.	0

10.7.2 Performance Counter 1 Control Register

Mnemonic: **PERF_1_CONTROL**

Address: (via MEMBASE-III):

Using shifted destination address $0x3_0000: 0x2078 + 0x4000*N$ ($N=0:3$)

Using shifted destination address $0x4_0000: 0x2078 + 0x4000*(N-4)$ ($N=4:7$)

Description:

Table 305 Performance Counter 1 Control Register Bits

Bit(s)	Access	Name	Description	Reset State																														
31	R/W	LCLK_TRIG	When triggered (written to logic 1), causes all performance counters to update. Self clears when done.	0																														
30	R/W	LCLK_CLEAR	This bit is reserved.	1																														
29	R/W	PERF_PRESET_TRIGGER	When triggered (written to logic 1), causes all performance counters to be preset to 0xFFFFFFFF0. Self clears when done.	0																														
[28:25]	—	UNUSED		XXXX																														
[24:20]	R/W	PERF1_EVENT_SELECT	<p>Indicates which event to count as follows:</p> <p>PERF1_EVENT_SELECT Encoding</p> <table> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>An SMP connection has reached the maximum allowed connection time.</td> </tr> <tr> <td>1</td> <td>An STP connection has reached the maximum allowed connection time.</td> </tr> <tr> <td>2</td> <td>An SSP connection has reached the maximum allowed connection time.</td> </tr> <tr> <td>3</td> <td>We transmitted SATA_R_ERR.</td> </tr> <tr> <td>4</td> <td>We received SATA_R_ERR.</td> </tr> <tr> <td>5</td> <td>We received a good STP frame.</td> </tr> <tr> <td>6</td> <td>We transmitted an STP frame.</td> </tr> <tr> <td>7</td> <td>We received a bad SMP frame and broke the connection.</td> </tr> <tr> <td>8</td> <td>We transmitted an SMP frame and received a break response.</td> </tr> <tr> <td>9</td> <td>We received a good SMP frame.</td> </tr> <tr> <td>10</td> <td>We transmitted a good SMP frame.</td> </tr> <tr> <td>11</td> <td>We transmitted a NAK (CRC_ERROR)</td> </tr> <tr> <td>12</td> <td>We received a NAK (CRC_ERROR) or an ACK/NAK timeout occurred.</td> </tr> <tr> <td>13</td> <td>We received a good SSP frame.</td> </tr> </tbody> </table>	Value	Definition	0	An SMP connection has reached the maximum allowed connection time.	1	An STP connection has reached the maximum allowed connection time.	2	An SSP connection has reached the maximum allowed connection time.	3	We transmitted SATA_R_ERR.	4	We received SATA_R_ERR.	5	We received a good STP frame.	6	We transmitted an STP frame.	7	We received a bad SMP frame and broke the connection.	8	We transmitted an SMP frame and received a break response.	9	We received a good SMP frame.	10	We transmitted a good SMP frame.	11	We transmitted a NAK (CRC_ERROR)	12	We received a NAK (CRC_ERROR) or an ACK/NAK timeout occurred.	13	We received a good SSP frame.	0 0000
Value	Definition																																	
0	An SMP connection has reached the maximum allowed connection time.																																	
1	An STP connection has reached the maximum allowed connection time.																																	
2	An SSP connection has reached the maximum allowed connection time.																																	
3	We transmitted SATA_R_ERR.																																	
4	We received SATA_R_ERR.																																	
5	We received a good STP frame.																																	
6	We transmitted an STP frame.																																	
7	We received a bad SMP frame and broke the connection.																																	
8	We transmitted an SMP frame and received a break response.																																	
9	We received a good SMP frame.																																	
10	We transmitted a good SMP frame.																																	
11	We transmitted a NAK (CRC_ERROR)																																	
12	We received a NAK (CRC_ERROR) or an ACK/NAK timeout occurred.																																	
13	We received a good SSP frame.																																	

Bit(s)	Access	Name	Description	Reset State
			14 We transmitted an SSP frame. 15 We detected a CRC error in a received data frame. 16 We detected an error in the received test pattern. 17 A break timeout occurred. 18 A connection was established. 19 AIP (WAITING_ON_CONNECTION) was received. 20 AIP (WAITING_ON_PARTIAL) was received. 21 We transmitted CREDIT_BLOCKED. 22 We received CREDIT_BLOCKED. 23 We transmitted a retry class open reject. 24 We received a retry class open reject. 25 We transmitted an abandon class open reject. 26 We received an abandon class open reject. 27 We initiated a break. 28 We received a break. 29 We detected a CRC error on a received address frame. 30 We detected a SAS error primitive Others Reserved	
[19:17]	R/W	PERF1_LINK_SELECT	Indicates which link to count events from	000
16	R/W	PERF1_SATURATE	When logic 0, performance counter saturates. When logic 1, performance counter rolls over.	0
[15:0]	R/W	PERF1_WINDOW	The period of the static window in 100 µs. 0 means the window is infinite. The window has an arbitrary starting point and restarts where the previous window finished. The performance counter is reset at the start of each window.	0000 0000 0000 0000

10.7.3 Performance Counter 1 Threshold Register

Mnemonic: **PERF_1_THRESHOLD**

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x207C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x207C + 0x4000*(N-4) (N=4:7)

Description:

Table 306 Performance Counter 1 Threshold Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	PERF1_THRESHOLD	This field records the threshold for the number of performance counter events that can be detected within a static (not sliding) window.	0000 0000 0000 0000 0000 0000 0000 0000

10.7.4 Performance Counter 1 Count Register

Mnemonic: **PERF_1_COUNT**

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x2080 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x2080 + 0x4000*(N-4) (N=4:7)

Description:

Table 307 Performance Counter 1 Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	PERF1_COUNT	Number of events detected. LCLK must be triggered for this register to update. See LCLK_CLEAR for details about clearing the counter. See PERF1_SATURATE for details about saturation.	0000 0000 0000 0000 0000 0000 0000 0000

10.7.5 Performance Counter 2 Control Register

Mnemonic: **PERF_2_CONTROL**

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x2084 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x2084 + 0x4000*(N-4) (N=4:7)

Description:

Table 308 Performance Counter 2 Control Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:25]	—	UNUSED		XXX XXXX
[24:20]	R/W	PERF2_EVENT_SELECT	Indicates which event to count. Encoding is the same as PERF1_EVENT_SELECT.	0 1111
[19:17]	R/W	PERF2_LINK_SELECT	Indicates which link to count events from	000
16	R/W	PERF2_SATURATE	When logic 0, performance counter saturates. When logic 1, performance counter rolls over.	0
[15:0]	R/W	PERF2_WINDOW	The period of the static window in 100 µs. 0 means the window is infinite. The window has an arbitrary starting point and restarts where the previous window finished. The performance counter is reset at the start of each window.	0000 0000 0000 0000

10.7.6 Performance Counter 2 Threshold Register

Mnemonic: **PERF_2_THRESHOLD**

Address: (via MEMBASE-III):

Using shifted destination address $0x3_0000: 0x2088 + 0x4000*N$ (N=0:3)

Using shifted destination address $0x4_0000: 0x2088 + 0x4000*(N-4)$ (N=4:7)

Description:

Table 309 Performance Counter 2 Threshold Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	PERF2_THRESHOLD	This field records the threshold for the number of performance counter events that can be detected within a static (not sliding) window.	0000 0000 0000 0000 0000 0000 0000 0000

10.7.7 Performance Counter 2 Count Register

Mnemonic: **PERF_2_COUNT**

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x208C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x208C + 0x4000*(N-4) (N=4:7)

Description:

Table 310 Perf 2 Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	PERF2_COUNT	Number of events detected. LCLK must be triggered for this register to update. See LCLK_CLEAR for details about clearing the counter. See PERF1_SATURATE for details about saturation.	0000 0000 0000 0000 0000 0000 0000 0000

10.7.8 Peak Detector Control Register

Mnemonic: **PEAK_DETECTOR_CONTROL**

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x2090 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x2090 + 0x4000*(N-4) (N=4:7)

Description:

Table 311 Peak Detector Control Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:6]	—	UNUSED		XX XXXX XXXX XXXX XXXX XXXX XXXX

Bit(s)	Access	Name	Description	Reset State
[5:4]	R/W	PEAK_EVENT_SELECT	Indicates which event to peak detect. PEAK_EVENT_SELECT Encoding 0: Maximum transmitted PBC 1: Maximum connection time, units are in 100 µs 2: Maximum transmitted AWT, units are in 1 µs until 0x7FFF, then in 1ms after that 3: Maximum arbitration wait time, units are in 1 µs until 0x7FFF, then in 1 ms after that	00
3	R/W	PEAK_CLEAR_TRIG	When set to logic 1, clears the peak detector. Automatically returns to 0 after a maximum of 10 clock cycles.	0
[2:0]	—	UNUSED		XXX

10.7.9 Peak Detector 1 Count Register

Mnemonic: PEAK_DETECTOR_1_COUNT

Address: (via MEMBASE-III):
 Using shifted destination address 0x3_0000: 0x2094 + 0x4000*N (N=0:3)
 Using shifted destination address 0x4_0000: 0x2094 + 0x4000*(N-4) (N=4:7)

Description:

Table 312 Peak Detector 1 Count Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R	PEAK_COUNT	Maximum value of the count detected	0000 0000 0000 0000 0000 0000 0000 0000

10.7.10 Peak Detector 1 Threshold Register

Mnemonic: PEAK_DETECTOR_1_THRESHOLD

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x2098 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x2098 + 0x4000*(N-4) (N=4:7)

Description:

Table 313 Peak Detector 1 Threshold Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	PEAK_THRESHOLD	PEAK_THR triggers when the count equals or exceeds this threshold.	0000 0000 0000 0000 0000 0000 0000 0000

10.7.11 Timer Enables Register

Mnemonic: TIMER_ENABLES

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x209C + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x209C + 0x4000*(N-4) (N=4:7)

Description:

Table 314 Timer Enables Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:29]	—	UNUSED		XXX
[28:21]	R/W	SMP_MAX_CONN_TIMER	<p>This timer specifies the maximum amount of time (in 100 µs units) an SMP connection may remain connected. 0 means timer is disabled. Note the SAS standard may require a per SMP command class timeout value. Firmware can implement this by setting up this register every time before establishing an SMP connection.</p> <p>The values for the SSP and STP connections are controlled by the MAX_CONNECT_TIME_SSP and MAX_CONNECT_TIME_STP fields in the OSSP Timer Control 0 register.</p>	0001 0100

Bit(s)	Access	Name	Description	Reset State
[20:15]	R/W	MAX_CONN_ENABLE	This enables the Maximum connect timer according to the connection type. The following connection types are supported starting from bit 5: SSP initiator, SSP Target, SMP initiator, SMP target, STP initiator, STP target.	11 1111
[14:9]	R/W	BUS_INACTIVITY_ENABLE	This enables the Bus Inactivity timer according to the connection type. The following connection types are supported starting from bit 5: SSP initiator, SSP Target, STP initiator, STP target., SMP initiator, SMP target	01 0100
[8:7]	R/W	STP_FRAME_ENABLE	STP_FRAME_ENABLE[1] allows the STPFrameTimer to timeout frame transmissions (from the point of sending an XRDY to receiving SYNCs). STP_FRAME_ENABLE[0] allows STPFrameTimer to timeout frame Receptions (from the point of receiving an XRDY to receiving a SYNC)	01
[6:0]	R/W	PROTOCOL_TIMER_DISABLE	This can be used to disable the SAS protocol timers for debug purposes. The following timeouts can be disabled starting from bit 0: identify, open, close, break, ACK/NAK, done, credit	000 0000

10.7.12 Rate Control Register

Mnemonic: RATE_CONTROL

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x20A8 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x20A8 + 0x4000*(N-4) (N=4:7)

Description:

Table 315 Rate Control Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	R/W	SSP_FRAME_RATE	The number of idle DWords between each SSP frame. 0 means no idle cycles.	0000 0000
[23:12]	R/W	STP_ALIGN_RATE	Align Insertion rate is 2 in every ALIGN_RATE+1 DWords. Default value results in standard compliant value of 2/256. This rate applies to out of STP connections.	0000 1111 1111

Bit(s)	Access	Name	Description	Reset State
[11:0]	R/W	ALIGN_RATE	Align Insertion rate is 2 in every ALIGN_RATE+1 DWords. Default value results in standard compliant value of 2/256. This rate applies to out of connection, SMP and SSP connections.	0000 1111 1111

10.7.13 Connection Time Register

Mnemonic: CONNECTION_TIME

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x20BC + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x20BC + 0x4000*(N-4) (N=4:7)

Description:

Table 316 Connection Time Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:29]	R/W	STP_FRAME_GAP	A minimum of the number of SATA_SYNCs specified by STP_FRAME_GAP is inserted between STP frame transmissions.	110
[28:16]	—	UNUSED		X XXXX XXXX XXXX
[15:0]	R	MAX_CON_TIMER_VALUE	This is the value of the internal maximum connection timer. The TIP_TRIG Register must be triggered for this to update. The connection time is (MaxConTimer - MAX_CON_TIMER_VALUE) * 100 µs period. For SMP connections, MaxConTimer refers to SMP_MAX_CONN_TIMER (Timer Enable Register). For SSP connections, MaxConTimer refers to MAX_CONNECT_SSP pin. For STP connections, MaxConTimer refers to MAX_CONNECT_STP. Note that MacConTimerValue count downwards from MaxConTimer.	0000 0000 0000 0000

10.7.14 General Purpose Register

Mnemonic: GENERAL_PURPOSE

Address: (via MEMBASE-III):

Using shifted destination address 0x3_0000: 0x20C0 + 0x4000*N (N=0:3)

Using shifted destination address 0x4_0000: 0x20C0 + 0x4000*(N-4) (N=4:7)

Description:

Table 317 General Purpose Register Bits

Bit(s)	Access	Name	Description	Reset State
31	R/W	SATA_DEVICE_MODE	When logic 1, the STP controller implement the device mode state machine if the PHY comes up in SATA mode. When logic 01, the STP controller implements the host mode state machine if the PHY comes up in SATA mode.	0
30	—	UNUSED		X
[29:26]	R/W	CHECK_MODE	These bits specify which test pattern to detect according to the table below. Scrambling must be turned off when using the JTPAT pattern since framing is needed to descramble properly. CHECK_MODE 0 checker off 1 check for CJTPAT and drop it 2 check for JTPAT and drop it 3 check for CJTPAT and pass it 4 check for JTPAT and pass it others reserved	0000
[25:22]	R/W	SYNC_GAIN_THR	The number of matches required to gain pattern sync. 0 is not a valid value for this field. The behavior of the detector is undefined when SYNC_GAIN_THR is set to 0.	1000
[21:18]	R/W	SYNC_LOSS_THR	The number of mismatches required to lose pattern sync. 0 is not a valid value for this field.	0100
17	—	UNUSED		X
16	R	PATTERN_SYNC	When logic 1, indicates the checker has detected the pattern specified by CHECK_MODE for more than SYNC_GAIN_THR consecutive DWords. When logic 0, indicates that the checker has detected a wrong pattern for SYNC_LOSS_THR consecutive DWords.	0
[15:4]	R/W	GPO	Spare RW bits	0001 0000 0000
[3:0]	R	GPI	Spare R bits	0000

10.8 SPC 8x6G Global Shared Memory Registers

The 32-bit BAR specified at PCI configuration address 0x20 (MEMBASE-III) contains the following chip registers used for configuring the GSM. The following subsections provide the descriptions for these registers.

The host must perform MEMBASE-III inbound window-shifting aligned on 64 KB as shown in Section 2.6.1, “[MEMBASE-III Inbound Window Shifting](#)”. The remaining space provided in the offset must be used as the offset for reading/writing the register using MEMBASE-III.

Table 318 Global Shared Memory Address Map

Offset	MEMBASE	Name
0x6A_0000: 0x80C0	MEMBASE-III	RB6 Access Register
0x70_0000: 0x0	MEMBASE-III	GSM Configuration and Reset Register
0x70_0000: 0x18	MEMBASE-III	GSM RAM ECC Double Bit Error Indication Register
0x70_0000: 0x38	MEMBASE-III	GSM Read Address Parity Check Enable Register
0x70_0000: 0x40	MEMBASE-III	GSM Write Address Parity Check Enable Register
0x70_0000: 0x48	MEMBASE-III	GSM Write Data Parity Check Enable Register
0x70_0000: 0x58	MEMBASE-III	GSM Read Address Parity Error Indication Register
0x70_0000: 0x60	MEMBASE-III	GSM Write Address Parity Error Indication Register
0x70_0000: 0x68	MEMBASE-III	GSM Write Data Parity Error Indication Register

10.8.1 GSM Configuration and Reset Register

Mnemonic: GSM_CFG_AND_RESET

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x0

Description: This register is used in the SPC 8x6G soft reset sequence. See [Table 368](#) in Section [11.4.2, “Soft Reset Recovery \(Normal Mode\)”](#) for details.

Table 319 GSM Configuration and Reset Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:15]	—	UNUSED		X XXXX XXXX XXXX XXXX
14	R/W	MST_XCBI_SW_RSTB	The Master XCBI Software Reset bit resets the Master XCBI logic. When result is logic: 1: The associated logic is not held reset. 0: The associated logic is held in reset.	1
13	R/W	COM_SLV_SW_RSTB	The Common Slave Software Reset bit resets the Common Slave logic. When result is logic: 1: The associated logic is not held reset. 0: The associated logic is held in reset.	1
12	R/W	QSSP_SW_RSTB	The QSSP I/O Context Software Reset bit resets the QSSP I/O Contextlogic. When result is logic: 1: The associated logic is not held reset. 0: The associated logic is held in reset.	1
11	R/W	RAAE_SW_RSTB	The RAAE I/O Context Software Reset bit resets the RAAE I/O Contextlogic. When result is logic: 1: The associated logic is not held reset. 0: The associated logic is held in reset.	1
10	—	UNUSED		X
9	R/W	RB_1_SW_RSTB	The Ring Buffer #1 Software Reset bit resets the Ring Buffer #1 logic. When result is logic: 1: The associated logic is not held reset. 0: The associated logic is held in reset.	1

Bit(s)	Access	Name	Description	Reset State
8	R/W	SM_SW_RSTB	The Shared Memory Software Reset bit resets the shared memory logic. When result is logic: 1: The associated logic is not held reset. 0: The associated logic is held in reset.	1
[7:4]	R/W	COHERENCY_GAP [3:0]	The Coherency Gap value controls the minimum spacing between RMW grant from the share memory arbiter. The Coherency Gap value is effective when COHERENCY_MODE = 1.	1000
3	R/W	COHERENCY_MODE	The Coherency Mode bit controls how the Shared Memory guarantees coherency for RMW operations. When set to logic: 1: Only one master at a time is granted a RMW operation. 0: The RMW operations are granted with a spacing that is controlled with the register value COHERENCY_GAP.	1
2	R/W	RB_WSTRB_ERRCHK_EN	The Ring Buffer Write Strobe Error Check Enable controls the checking of AXI WSTRB on the ring buffer interface. When set to logic: 1: Any value other then 0xFF on WSTRB is considered invalid and cause the GSM to report SLV ERR. 0: All WSTRB values are accepted on th.	0
1	R/W	RAAE_PORT2_EN	The RAAE Port 2 Enable bit controls access to the RAAE i/o context from Port #2. The RAAE i/o context is always available to AXI Slave Ports: 0,1 and 3 to 7. When set to logic: 1: The RAAE i/o context is accessible from slave AXI port #2. 0: The RAAE i/o context is not accessible from slave AXI port #2.	0

Bit(s)	Access	Name	Description	Reset State
0	R/W	GSM_WCI_MODE	<p>The GSM write to clear mode bit XOR with WCIMODE input controls the clearing of interrupts.</p> <p>When XOR result is logic:</p> <p>1: Interrupt registers are cleared when a logic one is written to the corresponding register bit.</p> <p>0: Interrupt registers are cleared upon reading the corresponding interrupt register.</p>	0

10.8.2 GSM RAM ECC Double Bit Error Indication Register

Mnemonic: RAM_ECC_DOUBLE_ERROR_INDICATOR

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x18

Description: This register is used in the soft reset sequence to indicate a GSM RAM ECC double bit error. This register is only used for debug purposes.

Table 320 RAM ECC Double Bit Error Indication Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:3]	—	UNUSED		X XXXX XXXX XXXX XXXX XXXX XXXX XXXX
2	R	ECC_ERR_DOUBLE_RB1_B_I0	<p>The ECC Double Error Indication bit is an event indication that reflects the detection of an uncorrectable double bit error.</p> <p>This bit indicates errors within the RING BUFFER #1 CONFIGURATION RAM.</p> <p>When read as logic:</p> <p>1: Single bit error was detected and corrected.</p> <p>0: No error was detected.</p>	0

Bit(s)	Access	Name	Description	Reset State
1	R	ECC_ERR_DOUBLE_RB1_A_I0	<p>The ECC Double Error Indication bit is an event indication that reflects the detection of an uncorrectable double bit error.</p> <p>This bit indicates errors within the RING BUFFER #1 POINTER RAM.</p> <p>When read as logic:</p> <ul style="list-style-type: none"> 1: Single bit error was detected and corrected. 0: No error was detected. 	0
0	R	ECC_ERR_DOUBLE_SERIALIZER_I0	<p>The ECC Double Error Indication bit is an event indication that reflects the detection of an uncorrectable double bit error.</p> <p>This bit indicates errors within the INT_SERIALIZER RAM.</p> <p>When read as logic:</p> <ul style="list-style-type: none"> 1: Single bit error was detected and corrected. 0: No error was detected. 	0

10.8.3 GSM Read Address Parity Check Enable Register

Mnemonic: READ_ADR_PARITY_CHK_EN

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x38

Description: This register is used in the SPC 8x6G soft reset sequence. See [Table 368](#) in Section [11.4.2, “Soft Reset Recovery \(Normal Mode\)”](#) for details.

Table 321 GSM Read Address Parity Check Enable Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:0]	R/W	ARADDR_PARITY_C HK_EN [23:0]	<p>The Read Address Parity Check Enable bit is a configuration bit for enabling parity checking on a per master port basis.</p> <p>When set to logic:</p> <p>1: Parity checking is enabled on the read address of the corresponding master port.</p> <p>0: Parity checking is disabled on the read address of the corresponding master port.</p>	1111 1111 1111 1111 1111 1111

10.8.4 GSM Write Address Parity Check Enable Register

Mnemonic: WRITE_ADR_PARITY_CHK_EN

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x40

Description: This register is used in the SPC 8x6G soft reset sequence. See [Table 368](#) in Section [11.4.2, “Soft Reset Recovery \(Normal Mode\)”](#) for details.

Table 322 GSM Write Address Parity Check Enable Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:0]	R/W	AWADDR_PARITY_CHK_EN [23:0]	<p>The Write Address Parity Enable bit is a configuration bit for enabling parity checking on a per master port basis.</p> <p>When set to logic:</p> <p>1: Parity checking is enabled on the write address of the corresponding master port.</p> <p>0: Parity checking is disabled on the write address of the corresponding master port.</p>	1111 1111 1111 1111 1111 1111

10.8.5 GSM Write Data Parity Check Enable Register

Mnemonic: WRITE_DATA_PARITY_CHK_EN

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x48

Description: This register is used in the SPC 8x6G soft reset sequence. See [Table 368](#) in Section [11.4.2, “Soft Reset Recovery \(Normal Mode\)”](#) for details.

Table 323 GSM Write Data Parity Check Enable Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:0]	R/W	WDATA_PARITY_CHK_EN [23:0]	<p>The Write Data Parity Enable bit is a configuration bit for enabling parity checking on a per master port basis.</p> <p>When set to logic:</p> <p>1: Parity checking is enabled on the write data of the corresponding master port.</p> <p>0: Parity checking is disabled on the write data of the corresponding master port.</p>	1111 1111 1111 1111 1111 1111

10.8.6 GSM Read Address Parity Error Indication Register

Mnemonic: READ_ADDR_PARITY_ERROR_INDICATOR

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x58

Description: This register is used in the soft reset sequence to indicate a GSM read address parity error. This register is only used for debug purposes.

Table 324 Read Address Parity Error Indication Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:20]	R	ARADDR_PARITY_I0 [23:0]	The Read Address Parity Indication bit is an event indicator that reflects a transition in the read address parity error of a particular master port. When read as logic: 1: A parity error has been detected. 0: No parity error has been detected.	0000 0000 0000 0000 0000 0000

10.8.7 GSM Write Address Parity Error Indication Register

Mnemonic: WRITE_ADDR_PARITY_ERROR_INDICATOR

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x60

Description: This register is used in the soft reset sequence to indicate a GSM write address parity error. This register is only used for debug purposes.

Table 325 Write Address Parity Error Indication Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:0]	R	AWADDR_PARITY_I0 [23:0]	The Write Address Parity Indication bit is an event indicator that reflects a transition in the write address parity error of a particular master port. When read as logic: 1: A parity error has been detected. 0: No parity error has been detected.	0000 0000 0000 0000 0000 0000

10.8.8 GSM Write Data Parity Error Indication Register

Mnemonic: WRITE_DATA_PARITY_ERROR_INDICATOR

Address: (via MEMBASE-III):
Using shifted destination address 0x70_0000: 0x68

Description: This register is used in the soft reset sequence to indicate a GSM write data parity error. This register is only used for debug purposes.

Table 326 Write Data Parity Error Indication Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:24]	—	UNUSED		XXXX XXXX
[23:0]	R	WDATA_PARITY_IO [23:0]	The Write Data Parity Indication bit is an event indicator that reflects a transition in the write data parity error of a particular master port. When read as logic: 1: A parity error has been detected. 0: No parity error has been detected.	0000 0000 0000 0000 0000 0000

10.8.9 RB6 Access Register

Mnemonic: RB6_ACCESS

Address: (via MEMBASE-III):
Using shifted destination address 0x6A_0000: 0x80C0

Description: This register is used in the SPC 8x6G soft reset sequence to cause an internal NMI to the SPC 8x6G processors. Use signature 0x00001234 to enable the SPC 8x6G NMI handler to put AAP1/IOP into wait state. See [Table 368](#) in Section 11.4.2, “Soft Reset Recovery (Normal Mode)” for details.

Table 327 RB6 Access Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	RB6 Access Data	The RB6 Access Data is used as a signature to trigger the NMI on the firmware side.	1111 1111 1111 1111 1111 1111 1111 1111

10.9 MBIC Registers

The 32-bit BAR specified at PCI configuration address 0x20 (MEMBASE-III) contains the following chip registers used for configuring the SPC 8x6G AAP1 and IOP in the MIPS34K Bridge and Interrupt Controller (MBIC) subsystem. The following subsections provide the descriptions for these registers.

The host must perform MEMBASE-III inbound window-shifting aligned on 64 KB as shown in Section 2.6.1, “[MEMBASE-III Inbound Window Shifting](#)”. The remaining space provided in the offset must be used as the offset for reading/writing the register using MEMBASE-III.

Table 328 MBIC Address Map

Offset	MEMBASE	Name
0x6_0000: 0x418	MEMBASE-III	NMI Enable VPE0 IOP Register
0x7_0000: 0x418	MEMBASE-III	NMI Enable VPE0 AAP1 Register

10.9.1 NMI Enable VPE0 AAP1 Register

Mnemonic: NMI_EN_VPE0_AAP1

Address: (via MEMBASE-III):
Using shifted destination address 0x 7_0000: 0x418

Description:

Table 329 NMI Enable VPE0 AAP1 Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	NMI_EN_VPE0	<p>This register is used to enable/disable non-maskable interrupt sources from generating an NMI interrupt to the processor (Virtual Processor 0)</p> <p>When set to logic:</p> <ul style="list-style-type: none"> 1: The corresponding NMI source is enabled. 0: The corresponding NMI source is masked. <p>The NMI sources driving the interrupt controller are shown in Table 330.</p>	0000 0000 0000 1000 0000 0000 0001 0000

Table 330 MBIC EXT_NMI[31:0] Sources

Bit	AAP1	IOP
0	External NMI	External NMI
1	INT[0] = CPU1 Fatal Memory Error Interrupt	INT[0] = CPU1 Fatal Memory Error Interrupt
2	INT[8] = CPU2 Fatal Memory Error Interrupt	INT[8] = CPU2 Fatal Memory Error Interrupt
3	INT[16] = CPU3 Fatal Memory Error Interrupt	INT[16] = CPU3 Fatal Memory Error Interrupt
4	INT[3] = CPU1 Watchdog Timer Interrupt	INT[19] = CPU1 Watchdog Timer Interrupt
5	PCIe AL Fatal	PCIe AL Fatal
6	PCIe AL Event	PCIe AL Event
7	GSM General	GSM General
8	BDMA Fatal	BDMA Fatal
9	MBIC1 Fatal	MBIC1 Fatal
10	SYS_RST[0]	SYS_RST[0]
11	SYS_RST[1]	SYS_RST[1]
12	Reserved	Reserved
13	GSM Fatal	GSM Fatal
14	Reserved	Reserved
15	MBIC2 Fatal	MBIC2 Fatal
16	MBIC3 Fatal	MBIC3 Fatal
17	OSSP Fatal	OSSP Fatal
18	Reserved	Reserved
19	RB 6	RB 6
20	TOP General	TOP General
21	MBIC General	MBIC General
22	PCSLM General	PCSLM General
23	OSSP General	OSSP General
24	OSSP Port General	OSSP Port General
25	Reserved	Reserved
26	PCIe General	PCIe General
27	PCIe AL General	PCIe AL General
28	PCIe PCS General	PCIe PCS General
31:29	Unused	Unused

10.9.2 NMI Enable VPE0 IOP Register

Mnemonic: NMI_EN_VPE0_IOP

Address: (via MEMBASE-III):
Using shifted destination address 0x 6_0000: 0x418

Description:

Table 331 NMI Enable VPE0 IOP Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:0]	R/W	NMI_EN_VPE0	<p>This register is used to enable/disable non-maskable interrupt sources from generating an NMI interrupt to the processor (Virtual Processor 0)</p> <p>When set to logic:</p> <ul style="list-style-type: none"> 1: The corresponding NMI source is enabled. 0: The corresponding NMI source is masked. <p>The NMI sources driving the interrupt controller are shown in Table 330.</p>	0000 0000 0000 1000 0000 0000 0000 0000

10.10 Miscellaneous Registers

Table 332 Miscellaneous Registers Address Map

Offset	MEMBASE	Name
0x 9_0000: 0x10C	MEMBASE-III	GPIO-0 Output Control Register

10.10.1 GPIO-0 Output Control Register

Mnemonic: GPIO_OUT_CTL_REG_1

Address: (via MEMBASE-III):
Using shifted destination address 0x 9_0000: 0x10C

Description:

Table 333 GPIO-0 Output Control Register Bits

Bit(s)	Access	Name	Description	Reset State
[31:2]	R/W	Reserved		
[1:0]	R/W	GPIO_OUT_CTL_0[1:0]	The GPIO_OUT_CTL_0 bits are used to configure how the output enable for GPIO[0] is driven. When set to logic: 00: GPIO[0] is tristated. 01: GPIO[0] is driven. 10: Reserved. 11: Reserved.	00

11 Error Information

The SPC 8x6G controller contains different error handling mechanisms (error detection, notification, and handling) for the different parts and sides of the device.

In general error sources are categorized into three different categories:

- PCIe errors (Section 11.1)
- SPC 8x6G device-specific errors (Sections 11.2, 11.3, and 11.4)
- SAS/SATA protocol specific errors (Sections 11.5, 11.6 and 11.7)

11.1 PCIe Errors

PCIe errors are reported by the SPC 8x6G controller to the Root Complex hardware through a message mechanism. These messages can be:

- Correctable
- Non-Fatal
- Fatal

Both Non-Fatal and Fatal PCIe errors require upper level software recovery in order for the system to continue or resume operation. Correctable PCIe errors can be corrected automatically by the PCIe hardware components involved in the transactions.

PCIe errors and messaging are described in Section 6.2 of the *PCI Express Base Specification 1.0a* and the associated *PCI Express Error Reporting ECN*. The SPC 8x6G implements the Advanced Error Reporting and Role Based Error Reporting mechanisms described in the *PCI Express Base Specification 1.0a*, and the associated *PCI Express Error Reporting ECN*.

11.1.1 Unsupported Requests

The SPC 8x6G is a single function endpoint device and is therefore not required to support the entire set of TLPs described in chapter 2 of the *PCI Express 1.0a Base Specification*. The following TLP types are not supported by the SPC 8x6G and cause an Unsupported Request completion response:

- MRdLk: Memory Read Lock
- CfgRd1: Configuration Read Type 1
- CfgWr1: Configuration Write Type 1
- Msg: Message (except Vendor Type 1 and Hot Plug Signaling)
- MsgD: Message with Data

Additionally, the SPC 8x6G responds with an Unsupported Request completion when the following requests are received:

- Configuration transaction targeting a non-existent function
- Memory or I/O request to an address not specified by any of the base address registers of any function
- Memory or I/O request of more than 4 bytes of DWord aligned data

Any Unsupported Request that requires a completion (Configuration Read/Write, I/O Read/Write, or Memory Read) is completed with an Unsupported Request in the status field of the completion. Memory writes to addresses outside of the defined Memory Base Address Registers are discarded without completion. In all cases, the URD (Unsupported Request Detected) bit in the Device Status Register is set.

11.2 Device Specific Fatal Errors

Device-specific fatal errors are conditions or actions that cause the SPC 8x6G controller operation to become unreliable.

Device-specific fatal errors are reported to the host driver software through the MSGU registers and interrupts, if enabled. When a fatal error occurs on the SPC 8x6G controller regardless of whether an interrupt is enabled or not, the device must be considered unreliable and either a [Soft Reset](#) (as described in Section 11.4.2) or a [Chip Reset](#) (as described in Section 11.4.4) must be done in order to resume reliable operation.

Fatal errors from each of the following SPC 8x6G subsystems (as described in Sections 1.3 and 1.4) will cause the SPC 8x6G to internally send an interrupt to firmware:

- Block Direct Memory Access (BDMA)
- Global Shared Memory (GSM)
- MIPS34K Bridge and Interrupt Controller (MBIC)
- PCIe Application Layer (PMIC)
- Octal SAS/SATA Port (OSSP)
- Hardened SAS/SATA Transport (HSST)
- Processor Complex System (PCS)

The SPC 8x6G firmware will read additional registers for more details about the error condition and will pass the error code to the host through the following MSGU registers:

- [Scratchpad 0 Register](#): When the [Scratchpad 1 Register](#) field [1:0], AAP_STATE, is set to 10b (error state), [Scratchpad 0 Register](#) reports details about the fatal error in the AAP1/MSGU.
- [Scratchpad 1 Register](#): Reports the general status and source for AAP1/MSGU errors.

- **Scratchpad 2 Register:** Reports the general status and source for IOP errors.
- **Scratchpad 3 Register:** When the **Scratchpad 2 Register** field [1:0], IOP_STATE, is set to 10b (error state), the **Scratchpad 3 Register** reports details about the fatal error in the IOP.

See Sections [10.2.5](#), [10.2.6](#), [10.2.7](#), and [10.2.8](#) for details about the Scratchpad registers.

If an interrupt to the host is available, the SPC 8x6G will generate one to notify the host of a fatal error condition. The Fatal Error Interrupt Enable (FERRIE) and Fatal Error Interrupt Vector (FERRIV) fields in the [MPI Main Configuration Table](#) (Section [5.2.1](#)) indicate the host interrupt setting to be used during a fatal error notification to the host.

The procedure to recover from a device-specific fatal error is described in Section [11.4](#), “[Device Specific Fatal Error Recovery Procedures](#)”.

The firmware ASSERT and watchdog timers are part of the fatal error source and will be reported in a manner similar to hardware subsystem fatal errors.

[Table 334](#) and [Table 335](#) show the fatal error source indications for the AAP1/MSGU and the IOP respectively.

In [Table 334](#), bits [31:8] indicate the source of the fatal error when bits [1:0], AAP_STATE, are set to 10b (error state). The **Scratchpad 0 Register** indicates further details about the error for the fatal error source shown in [Table 334](#). See the following subsections for more details about the contents of **Scratchpad 0 Register** for each error condition.

Table 334 Scratchpad 1 Register – AAP1/MSGU Status Indication for Fatal Error Case

Field	Name	Description
31	BDMA_ERR	BDMA module fatal error indicator.
30	GSM_ERR	GSM module fatal error indicator.
29	MBIC1_ERR	MBIC1 module fatal error indicator.
28	MBIC1_INTERNAL_SET0_ERR	MBIC1 module fatal error indicator.
27	MBIC1_INTERNAL_SET1_ERR	MBIC1 module fatal error indicator.
26	PMIC1_ERR	PMIC1 module fatal error indicator for AXI_MASTER_SLAVE_ERR.
25	PMIC2_ERR	PMIC2 module fatal error indicator for ERR - ECC_RAM_ERR.
24	PMIC_EVENT_ERR	PCIe AL event module fatal error indicator.
23	OSSP_ERR	OSSP module fatal error indicator.
22	Reserved	—
21	Reserved	—
20	HSST_ERR	HSST module fatal error indicator.
19	PCS_ERR	PCS module fatal error indicator.
18	PCIE_ERR	PCIE general fatal error indicator.

Field	Name	Description
17	Reserved	—
16	Reserved	—
15	ILA_FWLD_ERR	ILA loading firmware image error indicator.
14	FW_ASRT_ERR	Firmware Asserts fatal error indicator.
13	FW_WDG_ERR	Firmware Watchdog Timer fatal error indicator.
12	GEN_EXCEPTION_ERR	Firmware general exception indicator.
11	OTHER_CPU_ERR	Fatal error occurred on other CPU indicator.
10	MULTI_MOD_ERR	Fatal error indicator that could caused by one of a few different modules.
9	GENERAL_NMI_ERR	General NMI error indicator.
8	UNKNOWN_NMI_ERR	Unknown or multiple NMI error indicator.
[7:4]	Reserved	—
3	CPU_SOFT_RESET_RDY	AAP1 ready indicator for soft reset.
2	SFTRST_P_F	Toggled flag indicating Soft Reset progress.
[1:0]	AAP_STATE	Ready state for the AAP1/MSGU: 00: Power-on reset state. 01: Soft reset state. 10: Error state. 11: Ready state.

In Table 335, bits [31: 8] indicate the source of the fatal error when bits [1:0], IOP_STATE, are set to 10b (error state). The [Scratchpad 3 Register](#) reports further details of the error for the fatal error source shown in Table 335. See the following subsections for more details about the contents of [Scratchpad 3](#) for each error condition.

Table 335 Scratchpad 2 Register – IOP Status Indication for Fatal Error Case

Field	Name	Description
31	BDMA_ERR	BDMA module fatal error indicator.
30	GSM_ERR	GSM module fatal error indicator.
29	MBIC3_ERR	MBIC3 module fatal error indicator.
28	MBIC3_INTERNAL_SET0_ERR	MBIC3 module fatal error indicator.
27	MBIC3_INTERNAL_SET1_ERR	MBIC3 module fatal error indicator.
26	PMIC1_ERR	PMIC1 module fatal error indicator for AXI_MASTER_SLAVE_ERR.
25	PMIC2_ERR	PMIC2 module fatal error indicator for ERR - ECC_RAM_ERR.
24	PMIC_EVENT_ERR	PCIe AL event module fatal error indicator.
23	OSSP_ERR	OSSP module fatal error indicator.
22	Reserved	—
21	Reserved	—

Field	Name	Description
20	HSST_ERR	HSST module fatal error indicator.
19	PCS_ERR	PCS module fatal error indicator.
18	PCIE_ERR	PCIE general fatal error indicator.
17	Reserved	—
16	Reserved	—
15	ILA_FWLD_ERR	ILA loading firmware image error indicator.
14	FW_ASRT_ERR	FW Asserts fatal error indicator.
13	FW_WDG_ERR	FW Watch Dog Timer fatal error indicator.
12	GEN_EXCEPTION_ERR	Firmware general exception indicator.
11	OTHER_CPU_ERR	Fatal error occurred on other CPU indicator.
10	MULTI_MOD_ERR	Fatal error indicator that could caused by one of a few different modules.
9	GENERAL_NMI_ERR	General NMI error indicator.
8	UNKNOWN_NMI_ERR	Unknown or multiple NMI error indicator.
[7:4]	Reserved	—
3	CPU_SOFT_RESET_RDY	IOP ready indicator for soft reset.
2	HOST_SOFT_RESET_RDY	Ready indicator for host issuing a soft reset.
[1:0]	IOP_STATE	Ready state for the Input/Output Processing module: 00: Power-on reset state. 01: Soft reset state. 10: Error state. 11: Ready state.

For fatal error reporting, in addition to information provided in the [Scratchpad 0 Register](#), [Scratchpad 1 Register](#), [Scratchpad 2 Register](#) and [Scratchpad 3 Register](#), there are also register dumps available in the GSM that host may read. For details about register dumps, see Section [11.2.15, “Firmware Fatal Errors Register Dump”](#).

The offset of the memory dump for the MSGU is provided in the FERDOMSGU (Fatal Error Register Dump Offset for MSGU) field on DWord 0x1D of the Main Configuration Table (see Section [5.2.1](#)). The offset of the memory dump for the IOP is provided in the FERDOIOP (Fatal Error Register Dump Offset for IOP) field on DWord 0x1F of the Main Configuration Table (see Section [5.2.1](#)).

The format for the register dumps are ASCII. No parser is required.

11.2.1 Block Direct Memory Access (BDMA) Fatal Errors

This section describes the fatal errors caused by the BDMA subsystem as reported in bit [31], BDMA_ERR, in [Table 334](#) and [Table 335](#) for the [Scratchpad 1 Register](#) and the [Scratchpad 2 Register](#). Details of the error are reported in the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

There are three types of BDMA fatal errors that can be reported by the BDMA block. The type is represented by bits [31:30], BDMA_ERR_TYPE, described in [Table 336](#) below.

Table 336 BDMA Error Types

Field	Name	Description
[31:30]	BDMA_ERR_TYPE	<p>Source of the fatal error:</p> <p>00: BDMA_INTERRUPT1_EVENT: This exception is generated when the BDMA block detects a RAM ECC double-bit error during a:</p> <ul style="list-style-type: none"> • GSM-In/Out operation. • PCI operation. • SGL RAM operation. <p>01: BDMA_INTERRUPT2_EVENT: This exception is generated when the BDMA block detects a DMA channel error. This is caused by Advanced eXtensible Interface (AXI) response/parity errors on the BDMA core interface when:</p> <ul style="list-style-type: none"> • Fetching the Message Frame Address (MFA)/Message Frame (MF). • Writing back MFA. • Transferring DMA data. <p>10: BDMA_INTERRUPT3_EVENT: This exception is generated when the BDMA block detects an SGL Access error. This is triggered by an EOB error and AXI response/parity errors on read transactions performed by the SGL controller using either the PCIe, or GSM AXI when:</p> <ul style="list-style-type: none"> • Fetching BST • Fetching SGL entries <p>11: Reserved</p>
[29:0]	Error Detail	Detailed error information is defined in Table 337 , Table 338 and Table 339 below.

The details of the error in bits [29:0] depend on the type of BDMA error specified in bits [31:30], BDMA_ERR_TYPE. [Table 337](#), [Table 338](#) and [Table 339](#) below describes the BDMA errors for all three types defined in the BDMA_ERR_TYPE field ([Table 336](#)).

Table 337 BDMA Fatal Error Details – INTERRUPT 1 EVENT

Field	Name	Description
[31:30]	BDMA_ERR_TYPE	Source of the fatal error: 00: BDMA_INTERRUPT1_EVENT.
29	Reserved	—
[28:26]	SGL1_RAM_2BIT_ECC_ERR(x=2,1,0)	2-bit ECC RAM error in slice x of SGL1.
[25:24]	SGL2_RAM_2BIT_ECC_ERR(x=1,0)	2-bit ECC RAM error in slice x of SGL2.
[23:16]	Reserved	—
[15:12]	GSM_OUT_RAM_2BIT_ECC_ERR(x=3,2,1,0)	2-bit ECC RAM error in slice x of GSM out.
[11:8]	GSM_IN_RAM_2BIT_ECC_ERR(x=3,2,1,0)	2-bit ECC RAM error in slice x of GSM in.
[7:4]	PCI_RAM_2BIT_ECC_ERR(x=3,2,1,0)	2-bit ECC RAM error in slice x of PCI RAM.
[3:0]	Reserved	—

Table 338 BDMA Fatal Error Details - INTERRUPT 2 EVENT

Field	Name	Description
[31:30]	BDMA_ERR_TYPE	Source of the fatal error: 01: BDMA_INTERRUPT2_EVENT.
[29:20]	Reserved	—
19	DMA_GSM3_DEST_ERR_I2	DMA error from the GSM3 destination port.
18	DMA_GSM2_DEST_ERR_I2	DMA error from the GSM2 destination port.
17	DMA_GSM1_DEST_ERR_I2	DMA error from the GSM1 destination port.
16	DMA_GSM0_DEST_ERR_I2	DMA error from the GSM0 destination port.
[15:14]	Reserved	—
13	DMA_PCI_DEST_ERR_I2	DMA error from the PCIe destination port.
12	Reserved	—
11	DMA_GSM3_SRC_ERR_I2	DMA error from the GSM3 source port.
10	DMA_GSM2_SRC_ERR_I2	DMA error from the GSM2 source port.
9	DMA_GSM1_SRC_ERR_I2	DMA error from the GSM1 source port.
8	DMA_GSM0_SRC_ERR_I2	DMA error from the GSM0 source port.
7	GSM4_WR_ERR_I2	DMA write error from the GSM4 port.
6	GSM4_RD_ERR_I2	DMA read error from the GSM4 port.
5	DMA_PCI_SRC_ERR_I2	DMA error from the PCIe source port.
4	Reserved	—
3	DMA_DEST_PORT_ERR_I2	Error from the destination port.
2	DMA_SRC_PORT_ERR_I2	Error from the source port.
1	CORE_PARITY_ERR_I2	AXI response error on the BDMA Core interface.
0	AXI_RESPONSE_ERR_I2	AXI response error on BDMA Core interface.

Table 339 BDMA Fatal Error Details - INTERRUPT 3 EVENT

Field	Name	Description
[31:30]	BDMA_ERR_TYPE	Source of the fatal error: 10: BDMA_INTERRUPT3_EVENT.
[29:25]	Reserved	—
24	BST_PCI_PARITY_ERR	Parity error on BST PCI interface.
23	Reserved	—
22	BST_GSM_PARITY_ERR	Parity error on BST GSM interface.
21	BST_GSM_AXI_SLVERR	AXI response slave error on BST GSM interface.
20	BST_GSM_AXI_DECERR	AXI response decode error on BST GSM interface.
[19:18]	Reserved	—
17	BST_PCI_AXI_SLVERR	AXI response slave error on BST PCI interface.
16	BST_PCI_AXI_DECERR	AXI response decode error on BST PCI interface.
[15:10]	Reserved	—
9	SGL_GSM_PARITY_ERR_I3	Interrupt event due to parity error on the SGL GSM interface.
8	SGL_EOB_ERR	SGL cache reaching End of Buffer (EOB).
7	SGL_GSM_AXI_SLVERR_I3	Interrupt event due to AXI response slave error on the SGL GSM interface.
6	SGL_GSM_AXI_DECERR_I3	Interrupt event due to AXI response decode error on the SGL GSM interface.
[5:4]	Reserved	—
3	SGL_PCI_AXI_SLVERR	AXI response slave error on SGL PCI interface.
2	SGL_PCI_AXI_DECERR	AXI response decode error on SGL PCI interface.
1	Reserved	—
0	SGL_PCI_PARITY_ERR	Parity error on SGL PCI interface.

11.2.2 Global Shared Memory (GSM) Fatal Errors

This section describes the fatal errors caused by the GSM subsystem as reported in bit [30], **GSM_ERR**, of the [Scratchpad 1 Register](#) and the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#). The details of the errors are reported in the [Scratchpad 0 Register](#) or [Scratchpad 3 Registers](#) depending on whether the fatal errors are detected in the AAP1/MSGU or the IOP.

There are five sources of fatal errors that can be reported by the GSM block. These are listed in [31:29] of [Table 340](#), which describes the contents of the [Scratchpad 0 Register](#) or [Scratchpad 3 Registers](#) when a fatal GSM error is detected.

Table 340 GSM_ERR Types

Field	Name	Description
[31:29]	GSM_ERR	This field reports the source of the fatal error: 000: Double-Bit ECC Error Indications (DOUBLE_BIT_ECC_ERR). 001: Read Address Parity Errors (READ_ADDRESS_PARITY_ERR). 010: Write Address Parity Errors (WRITE_ADDRESS_PARITY_ERR). 011: Write Data Parity Errors (WRITE_DATA_PARITY_ERR). 100: Write Strobe Errors (WSTRB_ERR). 101: Reserved. 110: Reserved. 111: Reserved.
[28:0]	Error Detail	This field reports detailed information about the error. See details for each error below: Table 341: DOUBLE_BIT_ECC_ERR . Table 342: READ_ADDRESS_PARITY_ERR . Table 343: WRITE_ADDRESS_PARITY_ERR . Table 344: WRITE_DATA_PARITY_ERR . Table 345: WSTRB_ERR .

11.2.2.1 GSM Double-Bit ECC Error Details

This exception is generated when the GSM block detects a two-bit (double-bit) ECC error.

Table 341 GSM – DOUBLE_BIT_ECC_ERR Details

Field	Name	Description
[28:3]	Reserved	—
2	ECC_ERR_DOUBLE_RB1_B_i0.	This bit indicates errors within the RING BUFFER #1 CONFIGURATION RAM.
1	ECC_ERR_DOUBLE_RB1_A_i0	This bit indicates errors within the RING BUFFER #1 POINTER RAM.
0	ECC_ERR_DOUBLE_SERIALIZER_i0	This bit indicates errors within the INT_SERIALIZER.

11.2.2.2 GSM – Read Address Parity Error Details

This exception is generated when the GSM block detects a parity error on a read address operation.

Table 342 GSM – READ_ADDRESS_PARITY_ERR Details

Field	Name	Description
[28:24]	Reserved	—

Field	Name	Description
[23:0]	ARADDR_PARITY_i0 [23:0]	The Read Address Parity Indication bit is an event indicator that reflects a transition in the read address parity error of a particular master port.

11.2.2.3 GSM – Write Address Parity Error Details

This exception is generated when the GSM block detects a parity error on a write address operation.

Table 343 GSM – WRITE_ADDRESS_PARITY_ERR Details

Field	Field	Description
[28:24]	Reserved	—
[23:0]	AWADDR_PARITY_i0 [23:0]	The Write Address Parity Indication bit is an event indicator that reflects a transition in the write address parity error of a particular master port.

11.2.2.4 GSM – Write Data Parity Error Details

This exception is generated when the GSM block detects a parity error on a write data operation.

Table 344 GSM - WRITE_DATA_PARITY_ERR Details

Field	Name	Description
[28:24]	Reserved	—
[23:0]	WDATA_PARITY_i0 [23:0]	The Write Data Parity Indication bit is an event indicator that reflects a transition in the write data parity error of a particular master port.

11.2.2.5 GSM – Write Strobe Error Details

This exception is generated when the GSM block detects an unsupported WSTRB value while accessing the Ring Buffers.

Table 345 GSM – WSTRB_ERR Details

Field	Name	Description
[28:24]	Reserved	—

Field	Name	Description
[23:0]	WSTRB_ERR_i3 [23:0]	<p>The WSTRB Error Indication bit is an event indicator that reflects the detection of a non-supported WSTRB value when accessing the ring buffers.</p> <p>The ring buffer and I/O context systems only support an all-ones value for WSTRB.</p> <p>The shared memory system only supports an all-ones value when RMW = 0.</p>

11.2.3 MIPS34K Bridge and Interrupt Controller (MBIC) Fatal Errors

This section describes the fatal errors caused by the MBIC subsystem as reported in bit [29], MBIC1_ERR, of the [Scratchpad 1 Register](#), and MBIC3_ERR, of the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. The details of the errors are presented in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

There are five types of MBIC fatal errors that can be reported by the MBIC block:

- GSM AXI Access errors
- MBIC XCBI (MXCBI) errors
- PCIe AXI Access errors
- Internal Error 0 (see Section [11.2.3.4](#))
- Internal Error 1 (see Section [11.2.3.5](#))

Bits [31:28] of [Table 346](#) describe the contents of the [Scratchpad 0 Register](#) or [Scratchpad 3 Register](#) when a fatal MBIC error is detected.

Table 346 MBIC_ERR Types

Field	Name	Description
[31:28]	MBIC_ERR_TYPE	<p>This field reports the source of the fatal error:</p> <p>0000: GSM AXI Access (GSM_AXI_ERR).</p> <p>0001: MXCBI (MXCBI_AXI_ERR).</p> <p>0010: PCIe AXI Access (PCIE_AXI_ERR).</p> <p>[0011 –1111] Reserved.</p>
[27:0]	Error Detail	<p>This field contains the error details for each of the following MBIC errors:</p> <p>Table 347: GSM_AXI_ERR</p> <p>Table 348: MXCBI_AXI_ERR</p> <p>Table 349: PCIE_AXI_ERR</p>

11.2.3.1 MBIC – GSM AXI Error Details

This exception is generated when the MBIC block detects one of the following errors on the GSM AXI.

Table 347 MBIC - GSM_AXI_ERR Details

Field	Name	Description
[27:4]	Reserved	—
[3:0]	Error Detail	This field reports the different types of GSM AXI errors: 0001: GSM Read Data Parity Check (GSM_RDATA_PAR_CHK) 0010: GSM Read Address Inject Parity Error (GSM_ARADDR_INJECT_PAR_ERR) 0100: GSM Write Data Inject Parity Error (GSM_WDATA_INJECT_PAR_ERR) 1000: GSM Write Address Inject Parity Error (GSM_AWADDR_INJECT_PAR_ERR)

11.2.3.2 MBIC – MXCBI AXI Error Details

This exception is generated when the MBIC block detects one of the following errors on the MBIC XCBI (MXCBI).

Table 348 MBIC - MXCBI_AXI_ERR Details

Field	Name	Description
[27:4]	Reserved	—
[3:0]	Error Detail	This field reports the different types of MXCBI AXI errors. 0001: MXCBI Read Data Parity Check (MXCBI_RDATA_PAR_CHK) 0010: MXCBI Read Address Inject Parity Error (MXCBI_ARADDR_INJECT_PAR_ERR) 0100: MXCBI Write Data Inject Parity Error (MXCBI_WDATA_INJECT_PAR_ERR) 1000: MXCBI Write Address Inject Parity Error (MXCBI_AWADDR_INJECT_PAR_ERR)

11.2.3.3 MBIC – PCIe AXI Error Details

This exception is generated when the MBIC block detects one of the following errors on the PCIe AXI.

Table 349 MBIC - PCIE_AXI_ERR Details

Field	Name	Description
[27:4]	Reserved	—

Field	Name	Description
[3:0]	Error Detail	<p>This field reports the different types of PCIe AXI errors:</p> <ul style="list-style-type: none"> 0001: PCIe Read Data Parity Check (PCIE_RDATA_PAR_CHK) 0010: PCIe Read Address Inject Parity Error (PCIE_ARADDR_INJECT_PAR_ERR) 0100: PCIe Write Data Inject Parity Error (PCIE_WDATA_INJECT_PAR_ERR) 1000: PCIe Write Address Inject Parity Error (PCIE_AWADDR_INJECT_PAR_ERR)

11.2.3.4 MBIC – Internal Set 0 Error Details

This section describes the fatal error caused by the MBIC subsystem as reported in bit [28], **MBIC1_INTERNAL_SET0_ERR**, of the [Scratchpad 1 Register](#) and, **MBIC3_INTERNAL_SET0_ERR**, of the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

This exception is generated when the MBIC block detects various interrupt request bits that are associated with internal MBIC generated events.

Table 350 MBIC - INTERNAL_SET0_ERR Details

Field	Name	Description
[31:20]	Reserved	—
[19:16]	ERR_TAG	This field contains the value of OCP_MTagID[3:0] for the OCP transaction that resulted in an error. Note that in the case of writes, ERR_TAG represents the posted write buffer number. (All write transactions use a fixed OCP tag of 4'h7).
[15:8]	ERR_BYTEN	This field contains the value of OC_MByteEn for the OCP transaction that resulted in an error.
[7:4]	ERR_DEST	Indicates the destination of the OCP transaction that resulted in a failure: <ul style="list-style-type: none"> 000: Internal Registers. 001: GSM Shared Memory (Buffered). 002: GSM Shared Memory (Bypass Page Buffers). 003: GSM Queue Space. 004: GSM XCBI1 Region. 005: GSM XCBI2 Region. 006: Page Buffer Manager. 007: MXCBI1 Region. 008: MXCBI2 Region. 009: PMIC Region (PCIe AXI port). 00A: ESPRAM (for AAP1).

Field	Name	Description
3	ERR_BURST	ERR_BURST is set if the OCP transaction that resulted in a failure was a burst transaction.
2	ERR_RW	This field indicates the type (Read/Write) of an OCP transaction that resulted in a failure: 0: Indicates a Read transaction. 1: Indicates a Write transaction.
[1:0]	Reserved	—

11.2.3.5 MBIC – Internal Set 1 Error Details

This section describes the fatal error caused by the MBIC subsystem as reported in bit [27], MBIC1_INTERNAL_SET1_ERR, of the [Scratchpad 1 Register](#) and, MBIC3_INTERNAL_SET1_ERR, of the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

This exception is generated when the MBIC block detects various interrupt request bits that are associated with internal MBIC generated events.

Table 351 MBIC - INTERNAL_SET1_ERR Details

Field	Name	Description
[31:30]	Reserved	—
29	PMIC_RD_RESP_ERR_I2	Response error from the PMIC (PCIe) AXI interface for a read transaction to the PMIC address region.
28	MXCBI2_RD_RESP_ERR_I2	Response error from the MXCBI AXI interface for a read transaction to the MXCBI2 address region.
27	MXCBI1_RD_RESP_ERR_I2	Response error from the MXCBI AXI interface for a read transaction to the MXCBI1 address region.
26	PB_RD_RESP_ERR_I2	Response error from the Page Buffer for a read transaction to the Page Buffer register space.
25	GSM_XCBI2_RD_RESP_ERR_I2	Response error from the GSM AXI interface for a read transaction to the GSM_XCBI2 region.
24	GSM_XCBI1_RD_RESP_ERR_I2	Response error from the GSM AXI interface for a read transaction to the GSM_XCBI1 region.
23	GSM_Q_RD_RESP_ERR_I2	Response error from the GSM AXI interface for a read transaction to the GSM Queue space.
22	GSM_SM_RD_RESP_ERR_I2	Response error from the GSM AXI interface for a read transaction to the GSM_SM (Shared Memory unbuffered) space.
21	GSM_SMB_RD_RESP_ERR_I2	Response error from the GSM AXI interface for a read transaction to the GSM_SMB (Shared Memory Buffered) space.
20	REG_RD_RESP_ERR_I2	Response error from the XCBI slave module for a read transaction to an internal register.

Field	Name	Description
19	Reserved	—
18	PMIC_WR_RESP_ERR_I2	Response error from the PMIC (PCIe) AXI interface for a write transaction to the PMIC address region.
17	MXCBI2_WR_RESP_ERR_I2	Response error from the MXCBI AXI interface for a write transaction to the MXCBI2 address region.
16	MXCBI1_WR_RESP_ERR_I2	Response error from the MXCBI AXI interface for a write transaction to the MXCBI1 address region.
15	PB_WR_RESP_ERR_I2	Response error from the Page Buffer for a write transaction to the Page Buffer register space.
14	GSM_XCBI2_WR_RESP_ERR_I2	Response error from the GSM AXI interface for a write transaction to the GSM_XCBI2 Region.
13	GSM_XCBI1_WR_RESP_ERR_I2	Response error from the GSM AXI interface for a write transaction to the GSM_XCBI1 Region.
12	GSM_Q_WR_RESP_ERR_I2	Response error from the GSM AXI interface for a write transaction to the GSM Queue space.
11	GSM_SM_WR_RESP_ERR_I2	Response error from the GSM AXI interface for a write transaction to the GSM_SM (Shared Memory unbuffered) space.
10	GSM_SMB_WR_RESP_ERR_I2	Response error from the GSM AXI interface for a write transaction to the GSM_SMB (Shared Memory Buffered) space.
9	REG_WR_RESP_ERR_I2	Response error from the XCBI slave module for a write transaction to an internal register.
8	RDB_PARITY_ERR_I0	Parity Error detected by the Read Buffer Module. Note that this register bit is temporary and is subject to change in the next release.
7	PBM_PARITY_ERR_I0	Parity Error detected by Pager Buffer Module. Note that this register bit is temporary and is subject to change in the next release.
6	DMA_INT_I3	DMA operation complete interrupt.
[5:0]	Reserved	—

11.2.4 PMIC1 Error (AXI Master Slave)

This section describes the fatal error caused by the PMIC subsystem as reported in bit [26], PMIC1_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

Table 352 PMIC1_ERR – AXI_MASTER_SLAVE_ERR Details

Field	Name	Description
31	AL_SLV_WR_DATA_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Slave Write Data Port.

Field	Name	Description
30	AL_SLV_RD_DATA_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Slave Read Data Port.
29	AL_SLV_WR_ADDR_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Slave Write Address Port.
28	AL_SLV_RD_ADDR_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Slave Read Address Port.
27	AL_MSTR_WR_DATA_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Master Write Data Port.
26	AL_MSTR_RD_DATA_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Master Read Data Port.
25	AL_MSTR_WR_ADDR_PAR_ERR_I6	A parity error is detected on the PCIe Application Layer AXI Master Write Address Port.
24	AL_MSTR_RD_ADDR_PAR_ERR_I6	A parity error is detected on the PCIE Application Layer AXI Master Read Address Port.
23	AL_SLV_RD_RSP_SLV_ERR_I6	An SLV_ERR response is detected on the PCIE Application Layer AXI Slave Read Port.
22	AL_SLV_RD_RSP_DEC_ERR_I6	A DEC_ERR response is detected on the PCIE Application Layer AXI Slave Read Port.
21	AL_SLV_WR_RSP_SLV_ERR_I6	An SLV_ERR response is detected on the PCIE Application Layer AXI Slave Write Port.
20	AL_SLV_WR_RSP_DEC_ERR_I6	A DEC_ERR response is detected on the PCIE Application Layer AXI Slave Write Port.
19	AL_MSTR_RD_RSP_SLV_ERR_I6	A SLV_ERR response is detected on the PCIE Application Layer AXI Master Read Port.
18	AL_MSTR_RD_RSP_DEC_ERR_I6	A DEC_ERR response is detected on the PCIE Application Layer AXI Master Read Port.
17	AL_MSTR_WR_RSP_SLV_ERR_I6	An SLV_ERR response is detected on the PCIE Application Layer AXI Master Write Port.
16	AL_MSTR_WR_RSP_DEC_ERR_I6	A DEC_ERR response is detected on the PCIE Application Layer AXI Master Write Port.
15	PCIE_GM_CMPSR_LU_ERR_I6	PCIe-AXI Bridge Master Composer Lookup Error.
14	PCIE_RADMX_CMPSR_LU_ERR_I6	PCIe-AXI Bridge RADMX Composer Lookup Error.
13	PCIE_RADM_CPL_TIMEOUT_I6	PCIe Core RADM Completion Timeout Error.
12	PCIE_TRGT_CPL_TIMEOUT_I6	PCIe Core TRGT Completion Timeout Error.
11	PCIE_CFG_AER_RC_ERR_INT_I6	Error has occurred that cause bits in the Root Error Status Register of PCIe Core.
10	PCIE_CFG_AER_RC_ERR_MSI_I6	Error has occurred that caused bits in the Root Error Status Register of PCIe Core when MSI or MSIX is enabled.
9	PCIE_CFG_SYS_ERR_RC_I6	A root complex any device in the hierarchy reports ERR_COR, ERR_NONFATAL, or ERR_FATAL.
8	PCIE_RADM_CORRECTABLE_ERR_I6	An ERR_COR message is received.

Field	Name	Description
7	PCIE_RADM_NONFATAL_ERR_I6	An ERR_NONFATAL message is received.
6	PCIE_RADM_FATAL_ERR_I6	An ERR_FATAL message is received.
5	PCIE_RETRY_RAM_PAR_ERR_I6	A parity error is detected in the PCIe core Retry RAM.
4	PCIE_RETRY_SOT_PAR_ERR_I6	A parity error is detected in the PCIe core SOT Retry RAM.
3	PCIE_HDRQ_PAR_ERR_I6	A parity error is detected in the PCIe core Header Queue.
2	PCIE_DATAQ_PAR_ERR_I6	Interrupt is asserted (if enabled) when a parity error is detected in the PCIe core Data Queue.
1	AL_OB_TRANS_ERROR_I6	An outbound AXI transaction causes a fatal error.
0	AL_OB_WRID_MISS_I6	An outbound AXI write transaction response does not have the proper ID.

11.2.5 PMIC2 Error (ECC RAM)

This section describes the fatal error caused by the PMIC subsystem as reported in bit [25], PMIC2_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

Table 353 PMIC2_ERR – ECC_RAM_ERR Details

Field	Name	Description
[31:12]	Reserved	—
11	P_DATAQ_RAM_ECC_ERR_I5	An uncorrectable ECC error is detected in the Posted Data Queue RAM.
10	RADMX_COMP_RAM_ECC_ERR_3_I5	An uncorrectable ECC error is detected in the RADMX Composer 3 RAM.
9	RADMX_COMP_RAM_ECC_ERR_2_I5	An uncorrectable ECC error is detected in the RADMX Composer 2 RAM.
8	RADMX_COMP_RAM_ECC_ERR_1_I5	An uncorrectable ECC error is detected in the RADMX Composer 1 RAM.
7	RADMX_COMP_RAM_ECC_ERR_0_I5	An uncorrectable ECC error is detected in the RADMX Composer 0 RAM.
6	GM_COMP_RAM_ECC_ERR_I5	An uncorrectable ECC error is detected in the Generic Master Composer RAM.
5	RADMX_DECOMP_DATA_RAM_ECC_E_RR_I5	An uncorrectable ECC error is detected in the RADMX Decomposer Data RAM.
4	RADMX_TAG_ASYN FIFO_RAM_ECC_E_RR_I5	An uncorrectable ECC error is detected in the RADMX Tag RAM.
3	SLV_TAG_ASYN FIFO_RAM_ECC_ERR_I5	An uncorrectable ECC error is detected in the Slave Tag RAM.

Field	Name	Description
2	XADMX1_DECOMP_DATA_RAM_ECC_ERR_I5	An uncorrectable ECC error is detected in the XADMX1 Decomposer Data RAM.
1	XADMX0_DECOMP_DATA_RAM_ECC_ERR_I5	An uncorrectable ECC error is detected in the XADMX0 Decomposer Data RAM.
0	XDLH_RETRYRAM_RAM_ECC_ERR_I5	An uncorrectable ECC error is detected in the XDLH Retry RAM.

11.2.6 PMIC Event Error

This section describes the fatal error caused by the PMIC subsystem as reported in bit [24], PMIC_EVENT_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

Table 354 PMIC_EVENT_ERROR Details

Field	Name	Description
[31:27]	Reserved	—
26	ECC_COR_ERROR_INT_SUMMARY	Summary of the RAM ECC correctable error interrupt status register.
25	ECC_UNCOR_ERROR_INT_SUMMARY	Summary of the RAM ECC uncorrectable error interrupt status register.
24	PCI_ERROR_INT_SUMMARY	Summary of the PCIe error interrupt status register.
23:19	Reserved	—
18	RCVD_PME_TURNOFF_I4	A PM_PME_TURNOFF message received an interrupt.
17:6	Assigned for other usage	Root Complex Mode
5:4	Reserved	—
3	D3HOT_ENTRY_I4	The PM state machine transitions to a D3hot state interrupt.
2	D3HOT_D0_I4	The PM state machine transitions from a D3hot state to the D0 state interrupt.
1	LINK_REQ_RST_I4	Reports a link down condition interrupt.
0	TRAINING_RST_I4	Receiving a TS1/TS2 training sequence with the Hot Reset bit set.

11.2.7 Octal SAS/SATA Port (OSSP) Fatal Errors

This section describes the fatal error caused by the OSSP subsystem as reported in bit [23], OSSP_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

There are three types of fatal errors that can be reported by the OSSP block. These are listed in [31:29] of [Table 355](#), which describes the contents of the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) when a fatal OSSP error is detected

Table 355 OSSP_ERR Error Types

Field	Name	Description
[31:29]	OSSP_ERR_TYPE	Source of the fatal error: 000: GSM Decode Response Parity Error (GSM_DECODE_RESPONSE_PARITY_ERR). 001: FIFO ECC Error (FIFO_ECC_ERR). 010: HSST Global ECC Error (HSST_GLOBAL_ECC_ERR). 011: Reserved. 100: Reserved. 101: Reserved. 110: Reserved. 111: Reserved.
[28:0]	Error Detail	Detailed error information is defined in: Table 356: GSM_DECODE_RESPONSE_PARITY_ERR Table 357: FIFO_ECC_ERR Table 358: HSST_GLOBAL_ECC_ERR

11.2.7.1 OSSP – GSM Decode Response Parity Error Details

This exception is generated when the OSSP block detects a parity error in the GSM during operation.

Table 356 OSSP - GSM_DECODE_RESPONSE_PARITY_ERR Details

Field	Name	Description
[28:25]	PORT_NUMBER	The GSM AXI master Port number where error is being reported.
[24:10]	Reserved	—
9	WDATA_DEC_ERR_I	There is decoding error on the port. This represents an error response from GSM during a write transfer.
8	WDATA_SLV_ERR_I	There is slave error on the port. This represents an error response from GSM during a write transfer.

Field	Name	Description
[7:6]	Reserved	—
5	RDATA_DEC_ERR_I	There is decoding error on the port. This represents an error response from GSM during a read transfer.
4	RDATA_SLV_ERR_I	There is slave error on the port. This represents an error response from GSM during a read transfer.
[3:1]	Reserved	—
0	RDATA_PARITY_I	There is an uncorrectable parity error.

11.2.7.2 OSSP – FIFO ECC Error Details

This exception is generated when the OSSP block detects a parity error in one of its FIFOs during operation.

Table 357 OSSP - FIFO_ECC_ERR

Field	Name	Description
[28:25]	PHY_NUMBER	The PHY number where the error is being reported.
[24:14]	Reserved	—
[13:12]	TX_SYS_FIFO_ECC_ERR_I	TX system FIFO uncorrectable ECC error is detected.
[11: 6]	Reserved	—
[5:4]	RX_SYS_FIFO_ECC_ERR_I	RX system FIFO uncorrectable ECC error is detected.
[3:0]	Reserved	—

11.2.7.3 OSSP – HSST Global ECC Error Details

This exception is generated when the OSSP block detects an ECC error in one of its global RAMs during operation.

Table 358 OSSP - HSST_GLOBAL_ECC_ERR

Field	Name	Description
[28:06]	Reserved	—
[5:0]	GLOBAL_ECC_ERR_I1	A HSST Global ECC error is detected.

11.2.8 Hardened SAS/SATA Transport (HSST) Fatal Errors

This section describes the fatal error caused by the HSST subsystem as reported in bit [20], HSST_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

There are two types of HSST errors: Transport ECC errors and BDMA Timer Expired errors. [Table 359](#) lists the HSST error types.

Table 359 HSST_ERR Error Types

Field	Name	Description
[31:28]	HSST_ERR_TYPE	This field reports the source of the HSST fatal error: 0000: TRANSPORT_ECC_ERR_E1_P0 0001: TRANSPORT_ECC_ERR_E1_P1 0010: TRANSPORT_ECC_ERR_E1_P2 0011: TRANSPORT_ECC_ERR_E1_P3 0100: TRANSPORT_ECC_ERR_E1_P4 0101: TRANSPORT_ECC_ERR_E1_P5 0110: TRANSPORT_ECC_ERR_E1_P6 0111: TRANSPORT_ECC_ERR_E1_P7 1000: BDMA_TIMER_EXPIRED [1001:1111] Reserved
[27:0]	Error Detail	This field reports the detailed error information for each error type. See Table 360 (Transport ECC Errors) and Table 361 (BDMA Timer Expired Error) below.

11.2.8.1 HSST – Transport ECC Error E1 P[N] N:0-7 Details

This exception is generated when the HSST block detects an ECC error.

Table 360 HSST – TRANSPORT_ECC_ERR_E1_P[N] Details

Field	Name	Description
[27:24]	PHY_NUMBER	Zero-based PHY Identifier.
[23:8]	Reserved	—
[7:6]	WCS RAM	A RAM error in the Writable Control Storage block is detected.
[5:4]	FP DIF block RAM	A DIF RAM error in the Frame Processor block is detected.
[3:2]	FP header buffer RAM	A header buffer RAM error in the Frame Processor block is detected.
[1:0]	FP open profile RAM	An open profile RAM error in the Frame Processor block is detected.

11.2.8.2 HSST – BDMA Timer Expired Error Details

This exception is generated when the HSST block detects a BDMA timeout.

Table 361 HSST – BDMA_TIMER_EXPIRED Details

Field	Field	Description
[27:24]	PHY_NUMBER	The PHY number where the error is being reported.
[23:5]	Reserved	—
4	TX_BDMA_TIMER_IO	Interrupt bit indicating the TX BDMA timer has expired before the BDMA completion returns.
3	Reserved	—
2	BDMA_TIMER_IO	Interrupt bit indicating the RX BDMA timer has expired before the BDMA completion returns.
[1:0]	Reserved	—

11.2.9 Processor Complex System (PCS) Fatal Errors

This section describes the fatal error caused by the PCS subsystem as reported in bit [19], PCS_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

Table 362 PCS_ERR Details

Field	Name	Value
[31: 9]	Reserved	—
[8:6]	ARADDR_PARITY_ERR_I[2:0]	The Read Address Parity Error Indication bit is an event indicator that reflects a transition in the read address parity error of the corresponding interface.
[5:3]	AWADDR_PARITY_ERR_I [2:0]	The Write Address Parity Error Indication bit is an event indicator that reflects a transition in the write address parity error of a particular master port.
[2:0]	WDATA_PARITY_ERR_I [2:0]	The Write Data Parity Error Indication bit is an event indicator that reflects a transition in the write data parity error of a particular master port.

11.2.10 PCIe General Fatal Error

This section describes the fatal error caused by the PCIe core subsystem as reported in bit [18], PCIE_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#) described in Sections [10.2.6](#) and [10.2.7](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) described in Sections [10.2.5](#) and [10.2.8](#) respectively depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

Table 363 PCIE_ERR – PCIE - Uncorrectable Error Status

Field	Name	Description
[31:21]	Reserved	
20	UNSPRTD_REQ_ERR_STS	Unsupported Request Error Status
19	ECRC_ERR_STS	ECRC Error Status
18	MALFRMD_TLP_STS	Malformed TLP Status
17	RX_OVRFLW_STS	Receiver Overflow Status
16	UNEXPTD_COMPLTN_STS	Unexpected Completion Status
15	CMPLTR_ABRT_STS	Completer Abort Status
14	CMPLTN_TMOUT_STS	Completion Time Out Status
13	FLW_CNTRL_PRTCL_ERR_STS	Flow Control Protocol Error Status
12	POISND_TLP_STS	Poisoned TLP Status
[11:6]	Reserved	
5	SRPRS_DWN_ERR_STS	Surprise Down Error Status
4	DATA_LNK_PRTCL_ERR_STS	Data Link Protocol Error Status
[3:1]	Reserved	
0	UNDFND_PCIEXP	Undefined for PCI Express 1.1/2.0

11.2.11 ILA Image Loading Error

This section describes the fatal error caused by the ILA loading firmware image as reported in bit [15], ILA_FWLD_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). See [Table 334](#) and [Table 335](#) respectively. Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP.

Table 364 ILA Image Loading Error Details

Field	Name	Description
[31:20]	Reserved	
[19:12]	MOD ID	The firmware module ID: 01100010b - ILA module.
[11:0]	Error Detail	<p>Error details:</p> <ul style="list-style-type: none"> • 0x101: ILA_IMG_ISTRCRC_ERR – ISTR image has a CRC error • 0x110: ILA_IMG_ISTRBAD_ERR – ISTR image has a bad signature • 0x180: ILA_IMG_ISTRLEN_ERR – ISTR image's length is wrong • 0x201: ILA_IMG_AAP1CRC_ERR – AAP1 image has a CRC error • 0x280: ILA_IMG_AAP1LEN_ERR – AAP1 image's length is

Field	Name	Description
		<p>wrong</p> <ul style="list-style-type: none"> • 0x401: ILA_IMG_IOPCRC_ERR – IOP image has a CRC error • 0x480: ILA_IMG_IOPLEN_ERR – IOP image's length is wrong • 0x601: ILA_IMG_BOTH_AUTH_ERR – Firmware images on both firmware partitions (A and B) fails on authentification • 0x610: ILA_IMG_BOTHIMG_INVALID_ERR – Both flash partitions (A and B) do not have a set of valid firmware images • 0xE00: ILA_IMG_BOTHIMG_INACTIVE_ERR – Both firmware partitions (A and B) are not active • 0xFF5: ILA_PCIE_LINK_ERR – PCIe Link State error • 0xFF6: ILA_BOOTLOADER_FATAL_LED_ERR – Bootloader detects a fatal error • 0xFF7: ILA_SEEPROM_NODEVICE_ERR - Configuration SEEPROM does not exist • 0xFF8: ILA_SEEPROM_OPT_CRC_ERR - Configuration SEEPROM Optional Table CRC error • 0xFF9: ILA_SEEPROM_OPT_ERR - Configuration SEEPROM Optional Table Setting error • 0xFFA: ILA_SEEPROM_CFG_CRC_ERR - Configuration SEEPROM Configuration Table CRC error • 0xFFB: ILA_SEEPROM_CAP_CRC_ERR - Configuration SEEPROM Capability Setting CRC error • 0xFFC: ILA_SEEPROM_DEVICE_ERR – The SEEPROM device access error • 0xFFD: ILA_IMGLOADING_ERR – Error occurs while SPC 8x6G is loading firmware images • 0FFE: ILA_DEVICE_ERR – The device ID does not match with the SPC 8x6G device ID • 0FFF: ILA_HDA_NOSUPPORT_ERR – The ILA does not support HDA mode

11.2.12 Firmware Assert Fatal Error

This section describes the fatal error caused by firmware assertion as reported in bit [14], FW_ASRT_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP. See Sections [10.2.5](#), [10.2.6](#), [10.2.7](#), and [10.2.8](#) for details.

FW_ASRT_ERR information is also available when there is a fatal error triggered by a hardware interrupt. In this case, the current [Scratchpad 1 Register](#) and [Scratchpad 0 Register](#) for the AAP1 or [Scratchpad 2 Register](#) and [Scratchpad 3 Register](#) for the IOP are used to report the fatal error. The FW_ASRT_ERR information will be reported to the non-errored CPU for [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#). For example, if a fatal error occurs on the IOP, the [Scratchpad 2 Register](#) will contain the fatal error type, the [Scratchpad 3 Register](#) will contain the details for the error, and the [Scratchpad 0 Register](#) will contain the MOD_ID, ERROR_ID and LINE_NUMBER information.

Table 365 Firmware Assert Error Details

Field	Name	Description
[31:24]	MOD_ID	The firmware module ID code for the assert error.
[23:12]	ERROR_ID	The firmware module error ID, if any.
[11:0]	LINE_NUMBER	The error code line number.

11.2.13 Firmware Watchdog Timer Fatal Error

This section describes the fatal error caused by the firmware watchdog timer as reported in bit [13], FW_WDG_ERR, of the [Scratchpad 1 Register](#) the [Scratchpad 2 Register](#). Details of the error are reported in either the [Scratchpad 0 Register](#) or the [Scratchpad 3 Register](#) depending on whether the fatal error is detected in the AAP1/MSGU or the IOP. See Sections [10.2.5](#), [10.2.6](#), [10.2.7](#), and [10.2.8](#) for details.

The failed task that triggered the watchdog timer is reported in the non-errored CPU Scratchpad register. For example, for an AAP1 watchdog error, the [Scratchpad 1 Register](#) and the [Scratchpad 0 Register](#) report the fatal error and the failed task information is reported in the [Scratchpad 3 Register](#).

Table 366 Firmware Watchdog Timer Error Details

Field	Name	Description
[31:20]	Reserved	Reserved
[19:12]	MOD ID	The firmware module ID: 00110011b – Watchdog module ID (0x33).

Field	Name	Description
[11:0]	Error Detail (0x000)	0x001: WDG_ERR_FAIL 0x002: WDG_ERR_BLOCK_PERIOD_EXCEED_MAX 0x003: WDG_ERR_TIMEOUT_BELOW_MIN 0x004: WDG_ERR_UNEXPECTED_MSG 0x005: WDG_ERR_HW_WDG_TIMEOUT 0x006: WDG_ERR_MALLOC_FAILED 0x007: WDG_ERR_USER_PONG_FAILED 0x008: WDG_ERR_PONG_SENDER_UNEXPECTED 0x009: WDG_ERR_USER_PING_FAILED 0x00A: WDG_ERR_BUFFER_EXHAUSTED 0x00B: WDG_ERR_TMR_PING_FAILED

11.2.14 Firmware General Fatal Errors

The following general fatal errors do not provide error details to the host:

- OTHER_CPU_ERR – A fatal error that occurs on another CPU. This error condition indicates that the affected (live) CPU has intentionally entered a FREEZE state. When the offending CPU detects a fatal error, it will start the error handling process. When the process completes, the offending CPU will force the other live CPU to freeze by triggering an NMI interrupt. This is done to prevent the live CPU from processing the task further and to preserve the state of SPC 8x6G for debugging. Once the live CPU detects this condition, it will report an “OTHER_CPU_ERR”. There is no error detail information available for this CPU.
- MULTI_MOD_ERR – A fatal error interrupt generated by more than one module/source. This is a catch-all fatal error status. This case occurs when more than one source of fatal errors is detected while reading the NMI status register, or when there is no handler for top level fatal error interrupt. Because of the nature of this type of fatal error, there is no detailed error information available.
- GENERAL_NMI_ERR – An NMI fatal error generated by a host-initiated soft reset. The firmware just reports this error type to the scratchpad, no further error handling is required.
- UNKNOWN_NMI_ERR – An NMI fatal error generated by an undetermined source. This is a catch-all fatal error status. This case occurs when the firmware detects an exception from an undetermined source. Because of the nature of this type of fatal error, there is no detailed error information available.

The following general fatal error reports error details to the host through the Scratchpad registers:

- GEN_EXCEPTION_ERR – A fatal error caused by general exception. The offending code address is reported as error detail to host.

11.2.15 Firmware Fatal Errors Register Dump

When a device-specific fatal error occurs, the SPC 8x6G firmware puts normal operation on hold and forces the internal CPU into a freeze state. For debugging purposes, the SPC 8x6G firmware dumps certain register values first into the GSM and then into flash memory (if installed).

Note that not all fatal errors trigger a register value dump operation and not all register values are dumped. For those fatal errors that impact the GSM, the BDMA or general exceptions, a register dump may not occur. For example, if the GSM has a fatal error, nothing can be written to the GSM and a register dump will not occur. Similarly if a general exception occurs, any further operation will trigger a subsequent general exception and the register dump operation will become an infinite loop. In this case, the SPC 8x6G firmware may not trigger a register dump.

The register dump is in ASCII text format and no external parser is required. The contents of the register dump include numerous internal registers that SPC 8x6G firmware uses, but are not exposed in this document. The intended use of the register dump is to allow SPC 8x6G customers to capture register encoding and provide it to PMC-Sierra's support team for diagnosis.

The location of the register dump contents in the GSM is described in Section 5.2.1, "[MPI Main Configuration Table Fields](#)". DWords 0x1D to 0x20 contain information for the offset and length of the register dump for both the AAP1/MSGU and the IOP. The host can read the register dump by directly accessing this PCIe address space. The dump is cleared in the GSM following a soft reset, chip reset or, power-up.

To get the register dump contents from flash memory, use the [GET_NVMD_DATA Command](#) described in Section 7.27.

The following is a sample snapshot of a register dump.

Figure 50 Sample Register Dump Snapshot

```

PMC-SIERRA SPC FW version: 01080000 CPU ID: 0x02 (AAP1)

PCS_LM - BFC10000
[68]:[00000000] [70]:[00000000] [78]:[00000008] [80]:[00000008] [88]:[00000000]
[90]:[00000000] [98]:[00000000] [A0]:[000001FF] [B0]:[00000000]

BDMA - BF810000
[0]:[00000020] [10]:[00000000] [20]:[00000000] [24]:[00000000] [30]:[004D9480]
[34]:[00000000] [38]:[00010000] [3C]:[00000000] [40]:[00000000] [44]:[00000000]
[48]:[00000000] [4C]:[00000000] [60]:[00000000] [64]:[00000000] [68]:[00000000]
[6C]:[00000000] [70]:[00000000] [84]:[C1FF03FF] [88]:[00000000] [90]:[F00F3FFF]
[94]:[00000000] [9C]:[FF00FFFF] [A0]:[00000000] [A8]:[FF00FFFF] [AC]:[00000000]
[300]:[00000000]

MBIC - BCA00000
[0]:[04000002] [4]:[00000071] [104]:[000000F1] [108]:[00000000] [10C]:[00001000]

```

```
[114]:[00000000] [118]:[9FC23FD8] [11C]:[00000000] [120]:[000FF3FF]
[124]:[00000000] [128]:[000FFFC0] [134]:[00000001] [138]:[00000001]
[140]:[000000FA] [148]:[00000004] [14C]:[00000000] [180]:[00000000]
[184]:[00000000] [188]:[00000000] [18C]:[00000000] [190]:[00000000]
[198]:[1FC101E0] [19C]:[00070FA4] [400]:[00000000] [404]:[00000000]
[408]:[00000000] [410]:[FFFFFFFFFF] [414]:[00000000] [418]:[00000000]
[430]:[0000000F] [434]:[0000001F] [458]:[04C00154] [45C]:[00000000]
[460]:[04000050] [464]:[00000000] [468]:[00000000] [46C]:[00000000]
[470]:[00000000] [474]:[00000000] [4B8]:[00000003] [4BC]:[00000000]
[4C0]:[00000003] [4C4]:[00000000] [4D4]:[00000104] [4D8]:[00000000]
[4DC]:[00000000] [4E0]:[01000000] [4E4]:[00000000] [4E8]:[01555554]
[4EC]:[00002000] [4F0]:[09000800]
```

GSM - BFF00000

```
[0]:[00007B88] [8]:[00000000] [10]:[00000000] [18]:[00000000] [20]:[00000007]
[28]:[00000007] [38]:[00FFFFFF] [40]:[00FFFFFF] [48]:[00FFFFFF] [58]:[00000000]
[60]:[00000000] [68]:[00000000] [70]:[00FFFFFF] [78]:[00FFFFFF] [80]:[00FFFFFF]
[90]:[001C0000] [98]:[00000000] [A0]:[00000000] [A8]:[00FFFFFF] [C0]:[03FFFFFF]
[1800]:[00000000] [2000]:[00000000]
```

GSM FIFOs

id	free	post	comp
1	9FC71C00	0000	0000
42	0009C290	0000	0000
43	0009C490	0000	0000
44	0009C690	0000	0000
45	0009C790	0000	0000
46	0009C990	0000	0000
47	0009CB90	0000	0000
48	0009CC90	0000	0000
49	0009CD90	0000	0000
4A	0009CE90	0000	0000
4B	0009CF90	0000	0000
4C	0009D090	0000	0000
4D	0009D290	0000	0000
4E	0009D390	0000	0000
50	00085E20	0000	0000
51	00085F20	0000	0000
52	00086020	0000	0000
53	00086120	0000	0000
54	00086220	0000	0000
55	00086320	0000	0000
56	00086420	0000	0000
57	00086520	0000	0000
58	00086620	0000	0000
59	00086820	0000	0000
5A	00086A20	0000	0000
5B	00086C20	0000	0000
5C	00086E20	0000	0000
5D	00087020	0000	0000
5E	00087220	0000	0000
5F	00087420	0000	0000
60	00087620	0000	0000
61	000876D0	000876C2	0000
62	00087EC0	0000	0000
6B	00087F00	0000	0000
6C	00087F10	0000	0000
6D	00087F20	0000	0000
6E	00087F30	0000	0000

```

6F  00087F40 0000 0000
70  00087F50 0000 0000
71  00087F60 0000 0000
72  00087F70 0000 0000
73  00087F04 0000 0000
74  00087F14 0000 0000
75  00087F24 0000 0000
76  00087F34 0000 0000
77  00087F44 0000 0000
78  00087F54 0000 0000
79  00087F64 0000 0000
7A  00087F74 0000 0000
7C  00087F08 0000 0000
7D  00087F0C 0000 0000
7E  00453C00 0000 0000
7F  0041D100 0000 0000

```

OSSP - BF820000

```

[14]:[00000000] [28]:[00000000] [3C]:[00000000] [50]:[00000000] [64]:[00000000]
[78]:[00000000] [8C]:[00000000] [18]:[00000000] [2C]:[00000000] [40]:[00000000]
[54]:[00000000] [68]:[00000000] [7C]:[00000000] [90]:[00000000] [1C]:[00000000]
[30]:[00000000] [44]:[00000000] [58]:[00000000] [6C]:[00000000] [80]:[00000000]
[94]:[00000000] [A8]:[00000000] [20]:[00000000] [34]:[00000000] [48]:[00000000]
[5C]:[00000000] [70]:[00000000] [84]:[00000000] [98]:[00000000] [AC]:[00000000]
[B4]:[00000000] [120]:[00000000] [220]:[00000000] [320]:[00000000]
[420]:[00000000] [520]:[00000000] [620]:[00000000] [720]:[00000000]
[820]:[00000000] [124]:[20000000] [224]:[20000000] [324]:[20000000]
[424]:[20000000] [524]:[20000000] [624]:[20000000] [724]:[20000000]
[824]:[20000000]

```

HSST - 1F833000

```

[38]:[FFFF0000] [4038]:[FFFF0000] [8038]:[FFFF0000] [C038]:[FFFF0000]
[10038]:[FFFF0000] [14038]:[FFFF0000] [18038]:[FFFF0000] [1C038]:[FFFF0000]

[3C]:[FFFF0000] [403C]:[FFFF0000] [803C]:[FFFF0000] [C03C]:[FFFF0000]
[1003C]:[FFFF0000] [1403C]:[FFFF0000] [1803C]:[FFFF0000] [1C03C]:[FFFF0000]

[40]:[FFFF0000] [4040]:[FFFF0000] [8040]:[FFFF0000] [C040]:[FFFF0000]
[10040]:[FFFF0000] [14040]:[FFFF0000] [18040]:[FFFF0000] [1C040]:[FFFF0000]

[64]:[00000000] [4064]:[00000000] [8064]:[00000000] [C064]:[00000000]
[10064]:[00000000] [14064]:[00000000] [18064]:[00000000] [1C064]:[00000000]

[84]:[00000000] [4084]:[00000000] [8084]:[00000000] [C084]:[00000000]
[10084]:[00000000] [14084]:[00000000] [18084]:[00000000] [1C084]:[00000000]

[88]:[00000008] [4088]:[00000008] [8088]:[00000008] [C088]:[00000008]
[10088]:[00000008] [14088]:[00000008] [18088]:[00000008] [1C088]:[00000008]

[8C]:[00000000] [408C]:[00000000] [808C]:[00000000] [C08C]:[00000000]
[1008C]:[00000000] [1408C]:[00000000] [1808C]:[00000000] [1C08C]:[00000000]

[90]:[0000003F] [4090]:[0000003D] [8090]:[0000003D] [C090]:[0000001D]
[10090]:[0000003D] [14090]:[0000003D] [18090]:[0000001D] [1C090]:[0000001F]

[94]:[00000000] [4094]:[00000000] [8094]:[00000000] [C094]:[00000000]
[10094]:[00000000] [14094]:[00000000] [18094]:[00000000] [1C094]:[00000000]

```

[98]:[015C0397] [4098]:[015C0397] [8098]:[015C0397] [C098]:[015C0397]
 [10098]:[015C0397] [14098]:[015C0397] [18098]:[015C0397] [1C098]:[015C0397]

[C4]:[07800000] [40C4]:[07800000] [80C4]:[07800000] [C0C4]:[07800000]
 [100C4]:[07800000] [140C4]:[07800000] [180C4]:[07800000] [1C0C4]:[07800000]

[C8]:[00000000] [40C8]:[00000000] [80C8]:[00000000] [C0C8]:[00000000]
 [100C8]:[00000000] [140C8]:[00000000] [180C8]:[00000000] [1C0C8]:[00000000]

[CC]:[00000000] [40CC]:[00000000] [80CC]:[00000000] [C0CC]:[00000000]
 [100CC]:[00000000] [140CC]:[00000000] [180CC]:[00000000] [1C0CC]:[00000000]

[D0]:[00000000] [40D0]:[00000000] [80D0]:[00000000] [C0D0]:[00000000]
 [100D0]:[00000000] [140D0]:[00000000] [180D0]:[00000000] [1C0D0]:[00000000]

[D4]:[07000707] [40D4]:[07000707] [80D4]:[07000707] [C0D4]:[07000707]
 [100D4]:[07000707] [140D4]:[07000707] [180D4]:[07000707] [1C0D4]:[07000707]

PCIE APP - BF813000

[0]:[00040001] [4]:[00000000] [8]:[00000000] [10]:[0000D80D] [14]:[00000000]
 [18]:[00000010] [1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[00000000]
 [2C]:[00000000] [30]:[0000000F] [34]:[00000000] [38]:[00000000] [3C]:[00000000]
 [40]:[00040FCF] [44]:[00000000] [48]:[FFFFFFFF] [4C]:[00000000] [70]:[00000FFF]
 [74]:[00000000] [78]:[00000FF7] [7C]:[00000000] [320]:[00000000]
 [328]:[00000000] [32C]:[80000108] [330]:[00000000] [334]:[0000FFFB]
 [338]:[00000000] [350]:[00000000] [354]:[00000000] [358]:[00000000]
 [35C]:[00000000] [360]:[00033001] [364]:[00002002] [368]:[00000003]
 [36C]:[01010101] [370]:[00000000] [374]:[00000000] [378]:[00000000]
 [37C]:[00000000] [380]:[00000000] [384]:[00000000] [388]:[00000000]
 [38C]:[00000000]

PCIE PHY - BF814000

[0]:[00000000] [4]:[00000000] [C]:[00000001] [2C]:[00000000] [50]:[00000000]
 [54]:[00000101] [68]:[00000000] [13C]:[10001412] [1D0]:[00000000]
 [1F8]:[00000000] [1FC]:[00000003] [224]:[00000000] [230]:[00000000]
 [23C]:[10001412] [2D0]:[00000000] [2F8]:[00000000] [2FC]:[00000003]
 [324]:[00000000] [330]:[00000000] [33C]:[10001412] [3D0]:[00000000]
 [3F8]:[00000000] [3FC]:[00000000] [424]:[00000000] [430]:[00000000]
 [43C]:[10001412] [4D0]:[00000000] [4F8]:[00000000] [4FC]:[00000003]
 [524]:[00000000] [530]:[00003F00] [53C]:[10001412] [5D0]:[00000000]
 [5F8]:[00000000] [5FC]:[00000000] [624]:[00000000] [630]:[00003F00]
 [63C]:[10001412] [6D0]:[00000000] [6F8]:[00000000] [6FC]:[00000000]
 [724]:[00000000] [730]:[00003F00] [73C]:[10001412] [7D0]:[00000000]
 [7F8]:[00000000] [7FC]:[00000000] [824]:[00000000] [830]:[00003F00]
 [83C]:[10001412] [8D0]:[00000000] [8F8]:[00000000] [8FC]:[00000000]
 [900]:[00000000]

PCIE CORE - BF818000

[0]:[800111F8] [4]:[00100000] [8]:[01070005] [C]:[00000000] [10]:[00000004]
 [14]:[00000000] [18]:[00000004] [1C]:[00000000] [20]:[00000000] [24]:[00000000]
 [2C]:[00000000] [30]:[00000001] [3C]:[000001FF] [74]:[0C008701] [78]:[00002810]
 [80]:[10410000] [88]:[00000000] [98]:[00000000] [A0]:[00010002] [104]:[00000000]
 [110]:[00000000] [720]:[00000000] [73C]:[00000000]

SSPA - PHY: 0 - BF832000

[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
 [30]:[00000000] [74]:[00000000]

SSPA - PHY: 1 - BF836000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPA - PHY: 2 - BF83A000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPA - PHY: 3 - BF83E000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPA - PHY: 4 - BF842000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPA - PHY: 5 - BF846000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPA - PHY: 6 - BF84A000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPA - PHY: 7 - BF84E000
[1C]:[00000000] [20]:[00000000] [24]:[00000000] [28]:[18000000] [2C]:[00000000]
[30]:[00000000] [74]:[00000000]

SSPL - PHY: 0 - BF831000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1B00AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]

SSPL - PHY: 1 - BF835000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1700AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]

SSPL - PHY: 2 - BF839000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1300AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]

SSPL - PHY: 3 - BF83D000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1C00AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]

SSPL - PHY: 4 - BF841000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1000AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]

SSPL - PHY: 5 - BF845000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1D00AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]

SSPL - PHY: 6 - BF849000

```
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[2300AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]
```

```
SSPL - PHY: 7 - BF84D000
[0]:[008F7280] [4]:[57322408] [C]:[0000AA32] [10]:[00300000] [1C]:[00000100]
[20]:[00000000] [24]:[00000000] [28]:[00000000] [2C]:[00000000] [30]:[00000000]
[50]:[1800AA00] [70]:[5E7D8021] [74]:[00003F03] [78]:[00000000]
```

Notes

1. [Table 367](#) lists the internal registers reported in the register dump for diagnostic use by PMC-Sierra's Applications team.
2. The very first field in the register dump contains the header information such as PMC-Sierra's signature, version of the firmware, and the internal CPU ID.
3. The GSM ring buffer FIFO has three queues to present the status of a particular ring buffer. The register dump displays any non-empty ring buffer it detects. In the above example, ring buffer 6 is shown to have a CPU fatal error exception.
4. When bit 12, "GEN_EXCEPTION_ERR" is set in the Scratchpad 1 register as reported in [Table 334](#) and [Table 335](#), additional internal CPU register dump details are reported before other component register dumps as follows:

FAULT CODE (xxxxxxxx)

CPU REGISTERS:

```
Cause = 00000000    EPC = 00000000    Status = 00000000    BadVaddr = 00000000
Config = 00000000

GP      = 00000000    SP      = 00000000    FP      = 00000000    RA      =
00000000
r0(0)   = 00000000    r1(at)  = 00000000    r2(v0)  = 00000000    r3(v1)  =
00000000
r4(a0)   = 00000000    r5(a1)  = 00000000    r6(a2)  = 00000000    r7(a3)  =
00000000
r8(t0)   = 00000000    r9(t1)  = 00000000    r10(t2) = 00000000    r11(t3) =
00000000
r12(t4)  = 00000000    r13(t5) = 00000000    r14(t6) = 00000000    r15(t7) =
00000000
r16(s0)  = 00000000    r17(s1)  = 00000000    r18(s2)  = 00000000    r19(s3) =
00000000
r20(s4)  = 00000000    r21(s5)  = 00000000    r22(s6)  = 00000000    r23(s7) =
00000000
r24(24) = 00000000    r25(25) = 00000000    r26(k0) = 00000000    r27(k1) =
00000000
```

Table 367 Internal Registers Accessed For a Register Dump

Register Name	Offset
PCS_LM Registers	
PCSLM Interrupt Configuration Register	0x68
PCSLM Interrupt Status Register CPU-0	0x70
PCSLM Interrupt Status Register CPU-1	0x78
PCSLM Interrupt Status Register CPU-2	0x80
PCSLM Interrupt Enable CPU-0	0x88

Register Name	Offset
PCSLM Interrupt Enable CPU-1	0x90
PCSLM Interrupt Enable CPU-2	0x98
PCSLM Slave AXI Parity Check Enable	0xA0
PCSLM Slave AXI Error Indication	0xB0
BDMA Registers	
BDMA Configuration Register	0x00
BDMA Status Register	0x10
BDMA PCI BST Map Low Address Register	0x20
BDMA PCI BST Map High Address Register	0x24
BDMA GSM BST Map Low Address Register	0x30
BDMA GSM BST Map High Address Register	0x34
BDMA SGL Control	0x38
BDMA SGL Flush Control Register	0x3C
BDMA SGL Cache Debug Register	0x40
BDMA SGL AXI Debug Register	0x44
BDMA BDMA GSM AR Address Debug Register	0x48
BDMA BDMA GSM AW Address Debug Register	0x4C
BDMA PCI AR Address Debug1 Register	0x60
BDMA PCI AR Address Debug2 Register	0x64
BDMA PCI AW Address Debug1 Register	0x68
BDMA PCI AW Address Debug2 Register	0x6C
BDMA Interrupt Mask Register	0x70
BDMA Interrupt 3 Enable Register	0x84
BDMA Interrupt 3 Event Register	0x88
BDMA Interrupt 2 Enable Register	0x90
BDMA Interrupt 2 Event Register	0x94
BDMA Interrupt 1 Enable Register	0x9C
BDMA Interrupt 1 Event Register	0xA0
BDMA Interrupt 0 Enable	0xA8
BDMA Interrupt 0 Event	0xAC
BDMA DMA Request Queue Status Register	0x300
MBIC Registers	
MBIC CPU Configuration and Status Register	0x000
MBIC Configuration Register	0x004
MBIC GSM AXI Configuration Register	0x104
MBIC MXCBI AXI Configuration Register	0x108
MBIC PCIe AXI Configuration Register	0x10C
MBIC ESPRAM AXI Configuration Register	0x110
MBIC Scratchpad Register 0	0x114

Register Name	Offset
MBIC Scratchpad Register 1	0x118
MBIC MBIC Internal Interrupt Request 0 Register	0x11C
MBIC MBIC Internal Interrupt Enable 0 Register	0x120
MBIC MBIC Internal Interrupt Request 1 Register	0x124
MBIC MBIC Internal Interrupt Enable 1 Register	0x128
MBIC 34K Interrupt Request Register	0x134
MBIC 34K Interrupt Enable Register	0x138
MBIC Timer Block Configuration Register	0x140
MBIC Timer Interrupt Indication Register	0x148
MBIC Timer Interrupt Enable Register	0x14C
MBIC DMA Configuration Register 0	0x180
MBIC DMA Configuration Register 1	0x184
MBIC DMA Configuration Register 2	0x188
MBIC DMA Configuration Register 3	0x18C
MBIC DMA Configuration Register 4	0x190
MBIC Response Error Context Register 0	0x198
MBIC Response Error Context Register 1	0x19C
MBIC Software Interrupt VPE0 Register	0x400
MBIC Software Interrupt VPE1 Register	0x404
MBIC Software Event Register	0x408
MBIC Interrupt Enable VPE0 Register	0x410
MBIC Interrupt Enable VPE1 Register	0x414
MBIC NMI Enable VPE0 Register	0x418
MBIC Software Interrupt Priority Mask Register	0x430
MBIC Hardware Interrupt Priority Mask Register	0x434
MBIC Interrupt Summary Enable Register 0	0x458
MBIC Interrupt Summary Enable Register 1	0x45C
MBIC Interrupt Summary Enable Register 2	0x460
MBIC Interrupt Summary Enable Register 3	0x464
MBIC Interrupt Summary Enable Register 4	0x468
MBIC Interrupt Summary Enable Register 5	0x46C
MBIC Interrupt Summary Enable Register 6	0x470
MBIC Interrupt Summary Enable Register 7	0x474
MBIC Raw Interrupt Status VPE0 Register	0x4B8
MBIC Raw Interrupt Status VPE1 Register	0x4BC
MBIC Qualified interrupt Status VPE0 Register	0x4C0
MBIC Qualified interrupt Status VPE1 Register	0x4C4
MBIC Interrupt Summary Enable Register 0	0x4D4
MBIC Interrupt Summary Enable Register 1	0x4D8

Register Name	Offset
MBIC Interrupt Summary Enable Register 2	0x4DC
MBIC Interrupt Summary Enable Register 3	0x4E0
MBIC Interrupt Summary Enable Register 4	0x4E4
MBIC Interrupt Summary Enable Register 5	0x4E8
MBIC Interrupt Summary Enable Register 6	0x4EC
MBIC Interrupt Summary Enable Register 7	0x4F0
GSM Registers	
GSM Configuration and Reset Register	0x00
GSM PMC RAM BIST Control and Status Register	0x08
GSM RAM ECC Single Bit Error Indication Register	0x10
GSM RAM ECC Double Bit Error Indication Register	0x18
GSM RAM ECC Single Bit Error Interrupt Enable Register	0x20
GSM RAM ECC Double Bit Error Interrupt Enable Register	0x28
GSM Read Address Parity Check Enable Register	0x38
GSM Write Address Parity Check Enable Register	0x40
GSM Write Data Parity Check Enable Register	0x48
GSM Read Address Parity Error Indication Register	0x58
GSM Write Address Parity Error Indication Register	0x60
GSM Write Data Parity Error Indication Register	0x68
GSM Read Address Parity Error Interrupt Enable Register	0x70
GSM Write Address Parity Error Interrupt Enable Register	0x78
GSM Write Data Parity Error Interrupt Enable Register	0x80
GSM SRAM Refresh Interrupt Indication Register	0x90
GSM SRAM Refresh Interrupt Enable Register	0x98
GSM WSTRB Error Indication Register	0xA0
GSM WSTRB Error Interrupt Enable Register	0xA8
GSM 1T SRAM ECC Single Bit Error Indication Register	0xC0
GSM Queue Interrupt Indication 0 – 127 Register	0x1800
GSM Queue Interrupt Enable 0 – 127 Register	0x2000
OSSP Registers	
SSP GSM AXI master Port N Interrupt Enable Registers (N=0:6, base address + N*(0x14))	0x14
OSSP GSM AXI master Port N Interrupt Registers (N=0:6, base address + N*(0x14))	0x18
OSSP System FIFO N Interrupt Enable Registers (N=0:7, base address + N*(0x14))	0x1C
OSSP System FIFO N Interrupt Registers (N=0:7, base address + N*(0x14))	0x20
OSSP Global Interrupt Event Register	0xB4
OSSP - Global Configuration 1 Registers (N=0:7, base address + N*(0x100))	0x120
OSSP - PISO Configuration 1 Registers (N=0:7, base address + N*(0x100))	0x124
HSST Registers	
HSST(A) HSST DOC Abort Registers (A=0:7, base address + A*(0x4000))	0x38

Register Name	Offset
HSST(A) HSST SFO Abort Registers (A=0:7, base address + A*(0x4000))	0x3C
HSST(A) HSST HIP Abort Registers (A=0:7, base address + A*(0x4000))	0x40
HSST(A) HSST PM ITC0 Registers (A=0:7, base address + A*(0x4000))	0x64
HSST(A) HSST Interrupt 0 Enable Registers (A=0:7, base address + A*(0x4000))	0x84
HSST(A) HSST Interrupt 0 Registers (A=0:7, base address + A*(0x4000))	0x88
HSST(A) HSST Interrupt 1 Enable Registers (A=0:7, base address + A*(0x4000))	0x8C
HSST(A) HSST Interrupt 1 Registers (A=0:7, base address + A*(0x4000))	0x90
HSST(A) HSST Interrupt 2 Enable Registers (A=0:7, base address + A*(0x4000))	0x94
HSST(A) HSST Interrupt 2 Registers (A=0:7, base address + A*(0x4000))	0x98
HSST(A) HSST BDMA RX Close Config Registers (A=0:7, base address + A*(0x4000))	0xC4
HSST(A) HSST SM Debug Control Registers (A=0:7, base address + A*(0x4000))	0xC8
HSST(A) HSST SM Statemache 0 Registers (A=0:7, base address + A*(0x4000))	0xCC
HSST(A) HSST SM Statemache 1 Registers (A=0:7, base address + A*(0x4000))	0xD0
HSST(A) HSST Buffer Config Registers (A=0:7, base address + A*(0x4000))	0xD4
PCIe Application Registers	
PCIE Application Layer Configuration Control Register	0x00
PCIE Application Layer Inbound MSI Configuration Register 1	0x04
PCIE Application Layer Inbound MSI Configuration Register 2	0x08
PCIE Inbound WSM Configuration Register 1	0x10
PCIE Inbound WSM Configuration Register 2	0x14
PCIE Message Unit Configuration Register 1	0x18
PCIE Message Unit Configuration Register 2	0x1C
PCIE Inbound MSI Interrupt Enable Register	0x20
PCIE Inbound MSI Interrupt Status Register	0x24
PCIE Inbound WSM Interrupt Enable Register	0x28
PCIE Inbound WSM Interrupt Status Register	0x2C
PCIE Message Unit Inbound Doorbell Interrupt Enable Register	0x30
PCIE Message Unit Inbound Doorbell Interrupt Status Register	0x34
PCIE Message Unit Outbound Doorbell Clear Interrupt Enable Register	0x38
PCIE Message Unit Outbound Doorbell Clear Interrupt Status Register	0x3C
PCIE Event Interrupt Enable Register	0x40
PCIE Event Interrupt Register	0x44
PCIE Error Interrupt Enable Register	0x48
PCIE Error Interrupt Register	0x4C
PCIE ECC Uncorrectable Error Interrupt Enable Register	0x70
PCIE ECC Uncorrectable Error Interrupt Status Register	0x74
PCIE ECC Correctable Error Interrupt Enable Register	0x78
PCIE ECC Correctable Error Interrupt Status Register	0x7C
PCIE Message Unit Status Register	0x320

Register Name	Offset
PCIE Message Unit Host Interrupt Status Register	0x328
PCIE Message Unit Host Interrupt Mask Register	0x32C
PCIE Message Unit IOP Interrupt Status Register	0x330
PCIE Message Unit IOP Interrupt Mask Register	0x334
PCIE Message Unit Inbound Doorbell Clear Register	0x338
PCIE Message Unit Utility A Interrupt Counter/Timer	0x350
PCIE Message Unit Outbound Doorbell Register	0x354
PCIE Message Unit Outbound Doorbell Interrupt Mask Register	0x358
PCIE Message Unit Outbound Doorbell Auto Clear Register	0x35C
PCIE Message Unit Scratchpad Register 0	0x360
PCIE Message Unit Scratchpad Register 1	0x364
PCIE Message Unit Scratchpad Register 2	0x368
PCIE Message Unit Scratchpad Register 3	0x36C
PCIE Host Scratchpad Register 0	0x370
PCIE Host Scratchpad Register 1	0x374
PCIE Host Scratchpad Register 2	0x378
PCIE Host Scratchpad Register 3	0x37C
PCIE Host Scratchpad Register 4	0x380
PCIE Host Scratchpad Register 5	0x384
PCIE Host Scratchpad Register 6	0x388
PCIE Host Scratchpad Register 7	0x38C
PCIe PHY Registers	
PCIE_PCS_PMA Interrupt Enable Register	0x00
PCIE_PCS_PMA Interrupt Event Register	0x04
PCIE_PCS_PMA Test Control/Status Register	0x0C
PCIE_PCS_PMA PHY Control Register	0x2C
PCIE_PCS_PMA Fuse Program Status Register	0x50
PCIE_PCS_PMA Fuse Program Control Register	0x54
PCIE_PCS_PMA SMOD Configuration Register	0x68
PCIE_PCS_PMA Lane 0 DCR Configuration Register	0x13C
PCIE_PCS_PMA Lane 0 Diagnostics Configuration Register	0x1D0
PCIE_PCS_PMA Lane 0 Interrupt Enable Register	0x1F8
PCIE_PCS_PMA Lane 0 Interrupt Event Register	0x1FC
PCIE_PCS_PMA Lane 1 Receiver Status Register	0x224
PCIE_PCS_PMA Lane 1 DCR Status Register	0x230
PCIE_PCS_PMA Lane 1 DCR Configuration Register	0x23C
PCIE_PCS_PMA Lane 1 Diagnostics Configuration Register	0x2D0
PCIE_PCS_PMA Lane 1 Interrupt Enable Register	0x2F8
PCIE_PCS_PMA Lane 1 Interrupt Event Register	0x2FC

Register Name	Offset
PCIE_PCS_PMA Lane 2 Receiver Status Register	0x324
PCIE_PCS_PMA Lane 2 DCR Status Register	0x330
PCIE_PCS_PMA Lane 2 DCR Configuration Register	0x33C
PCIE_PCS_PMA Lane 2 Diagnostics Configuration Register	0x3D0
PCIE_PCS_PMA Lane 2 Interrupt Enable Register	0x3F8
PCIE_PCS_PMA Lane 2 Interrupt Event Register	0x3FC
PCIE_PCS_PMA Lane 3 Receiver Status Register	0x424
PCIE_PCS_PMA Lane 3 DCR Status Register	0x430
PCIE_PCS_PMA Lane 3 DCR Configuration Register	0x43C
PCIE_PCS_PMA Lane 3 Diagnostics Configuration Register	0x4D0
PCIE_PCS_PMA Lane 3 Interrupt Enable Register	0x4F8
PCIE_PCS_PMA Lane 3 Interrupt Event Register	0x4FC
PCIE_PCS_PMA Lane 4 Receiver Status Register	0x524
PCIE_PCS_PMA Lane 4 DCR Status Register	0x530
PCIE_PCS_PMA Lane 4 DCR Configuration Register	0x53C
PCIE_PCS_PMA Lane 4 Diagnostics Configuration Register	0x5D0
PCIE_PCS_PMA Lane 4 Interrupt Enable Register	0x5F8
PCIE_PCS_PMA Lane 4 Interrupt Event Register	0x5FC
PCIE_PCS_PMA Lane 5 Receiver Status Register	0x624
PCIE_PCS_PMA Lane 5 DCR Status Register	0x630
PCIE_PCS_PMA Lane 5 DCR Configuration Register	0x63C
PCIE_PCS_PMA Lane 5 Diagnostics Configuration Register	0x6D0
PCIE_PCS_PMA Lane 5 Interrupt Enable Register	0x6F8
PCIE_PCS_PMA Lane 5 Interrupt Event Register	0x6FC
PCIE_PCS_PMA Lane 6 Receiver Status Register	0x724
PCIE_PCS_PMA Lane 6 DCR Status Register	0x730
PCIE_PCS_PMA Lane 6 DCR Configuration Register	0x73C
PCIE_PCS_PMA Lane 6 Diagnostics Configuration Register	0x7D0
PCIE_PCS_PMA Lane 6 Interrupt Enable Register	0x7F8
PCIE_PCS_PMA Lane 6 Interrupt Event Register	0x7FC
PCIE_PCS_PMA Lane 7 Receiver Status Register	0x824
PCIE_PCS_PMA Lane 7 DCR Status Register	0x830
PCIE_PCS_PMA Lane 7 DCR Configuration Register	0x83C
PCIE_PCS_PMA Lane 7 Diagnostics Configuration Register	0x8D0
PCIE_PCS_PMA Lane 7 Interrupt Enable Register	0x8F8
PCIE_PCS_PMA Lane 7 Interrupt Event Register	0x8FC
PCIE_PCS_PMA PRBS/Test Pattern Sync Status Register	0x900
PCIe Core Registers	
PCIE Device ID and Vendor ID Register	0x00

Register Name	Offset
PCIE PCI Status/Command Register	0x04
PCIE Class Code/Revision ID Register	0x08
PCIE Cache Line Size Register	0x0C
PCIE Base Address 0 (Optional) Register	0x10
PCIE Base Address 1 Register	0x14
PCIE Base Address 2 (Optional) Register	0x18
PCIE Base Address 3 Register	0x1C
PCIE Base Address 4 (Optional) Register	0x20
PCIE Base Address 5 Register	0x24
PCIE Subsystem ID Subsystem Vendor ID Register	0x2C
PCIE Expansion ROM BASE Address Register	0x30
PCIE Interrupt Line Register	0x3C
PCIE Device Capabilities Register	0x74
PCIE Device Control Register	0x78
PCIE Link Control Register	0x80
PCIE Slot Control Register	0x88
PCIE Device Control 2 Register	0x98
PCIE Link Control 2 Register	0xA0
PCIE Uncorrectable Error Status Register	0x104
PCIE Correctable Error Status Register	0x110
PORT_LOGIC PHY Status Register	0x720
PORT_LOGIC Queue Status Register	0x73C
SSPA PHY Registers	
SSPA(A) Interrupt 0 Enable Register (A=0:7, base address + A*(0x4000))	0x1C
SSPA(A) Interrupt 0 Register (A=0:7, base address + A*(0x4000))	0x20
SSPA(A) Interrupt 1 Enable Register (A=0:7, base address + A*(0x4000))	0x24
SSPA(A) Interrupt 1 Register (A=0:7, base address + A*(0x4000))	0x28
SSPA(A) Interrupt 2 Enable Register (A=0:7, base address + A*(0x4000))	0x2C
SSPA(A) Interrupt 2 Register (A=0:7, base address + A*(0x4000))	0x30
SSPA(A) Interrupt Summary (A=0:7, base address + A*(0x4000))	0x74
SSPL PHY Registers	
SSPL_6G(A) Configuration 1 Register (A=0:7, base address + A*(0x4000))	0x00
SSPL_6G(A) Configuration 2 Register (A=0:7, base address + A*(0x4000))	0x04
SSPL_6G(A) Timer Configuration Register (A=0:7, base address + A*(0x4000))	0x0C
SSPL_6G(A) Connection Status Register (A=0:7, base address + A*(0x4000))	0x10
SSPL_6G(A) LMS Override Control Register (A=0:7, base address + A*(0x4000))	0x1C
SSPL_6G(A) Interrupt Enable 1 Register (A=0:7, base address + A*(0x4000))	0x20
SSPL_6G(A) Interrupt Event 1 Register (A=0:7, base address + A*(0x4000))	0x24
SSPL_6G(A) Interrupt Status 1 Register (A=0:7, base address + A*(0x4000))	0x28

Register Name	Offset
SSPL_6G(A) Interrupt Values 1 Register (A=0:7, base address + A*(0x4000))	0x2C
SSPL_6G(A) Counter Configuration Register (A=0:7, base address + A*(0x4000))	0x30
SSPL_6G(A) Diagnostics Configuration Register (A=0:7, base address + A*(0x4000))	0x50
SSPL_6G(A) SAS2 Configuration Register (A=0:7, base address + A*(0x4000))	0x70
SSPL_6G(A) SAS2 Settings (Local) Register (A=0:7, base address + A*(0x4000))	0x74
SSPL_6G(A) SAS2 Settings Remote) Register (A=0:7, base address + A*(0x4000))	0x78

11.3 Device Specific Recoverable/Correctable Errors

Notifications of recoverable errors are not immediately sent to the host. Instead, a set of counters are defined in the [MPI General Status Table](#) described in Section 5.2.2. These counters are the RERRINFOx fields of the [MPI General Status Table](#) (GST).

This feature is not currently implemented. It is planned for a future enhancement of the SPC 8x6G.

11.4 Device Specific Fatal Error Recovery Procedures

This section describes some of the options for recovering from device specific fatal errors as described in Section 11.2, “Device Specific Fatal Errors”.

There are currently two reset options available to recover from fatal errors:

- Soft Reset (normal mode and HDA mode)
- Chip Reset

In most cases, SPC 8x6G firmware will be able to report the fatal error code to the host through the MSGU registers. (For details, see Section 11.2, “Device Specific Fatal Errors”. Depending on the error code and the error source, the host can decide to either apply the soft reset or the chip reset recovery procedure.

Note that there are cases where only a chip reset is a viable option. For example, a chip reset is required when the PCIe subsystem has a fatal error, or when the SPC 8x6G firmware is not able to report a fatal error to the host because of the error itself. For the latter case, where the fatal error prevents the SPC 8x6G firmware from reporting details, the host should have a timeout mechanism to detect if the device is still “alive”. The host may query the MSGUTCNT and IOPTCNT fields in [MPI General Status Table](#) to see if the heartbeat timer counter is progressing. (See Section 5.2.2, “[MPI General Status Table Fields](#)”.)

The subsections below describe the reset properties of and steps needed to perform the particular reset.

11.4.1 Chip Status Determination Before Soft-Reset

Before the host issues the Soft Reset Recovery (Normal Mode or HDA Mode), it must determine the chip's current status from the host side.

The host can determine the chip's current status by the reading back the HDA response field RSP_CODE at byte offset 28:29 of the response block.

If the RSP_CODE is HDAR_IDLE (0x8002), the chip is in HDA mode, which can be caused by any of the four situations described in Section 3.21.1. In this case, the host should check the Scratchpad Registers described in Section 11.2 to determine the possible fatal errors that occurred during bootup. The host may choose to either:

- exit with failure or
- go to the step 2 of the Soft Reset Recovery (HDA mode) if there no fatal error occurred

If the RSP_CODE is not HDAR_IDLE(0x8002), the host can follow the full steps in the Soft Reset Recovery (Normal Mode or HDA mode) in Section 11.4.2 and Section 11.4.3.

11.4.2 Soft Reset Recovery (Normal Mode)

A soft reset is performed to recover the device from a specific fatal error. The term normal mode refers to the non-HDA mode. A soft reset cannot be performed for PCIe subsystem fatal errors. The soft reset procedure preserves the SPC 8x6G PCIe subsystem. As a result, the host will not see disruptions or changes in the host's PCIe Configuration Space.

At a high level, the host initiates a soft reset by resetting the SPC 8x6G's internal processors and the BDMA subsystem using the [SPC Reset Register](#)'s reset bits as described in Section 10.3.7. Prior to triggering the reset bits, the host must mark the soft reset signature in the [Host Scratchpad 0 Register](#) (see Section 10.2.9) in order to inform the SPC 8x6G during firmware reset booting that the soft reset path is to be used. When the soft reset signature is detected during device boot, the SPC 8x6G boot ROM and firmware initialization process will bypass the PCIe subsystem initialization.

Host-to-SPC 8x6G soft reset synchronization is defined using the [Scratchpad 1 Register](#) (Section 10.2.6) and the [Scratchpad 2 Register](#) (Section 10.2.7). The host is required to follow the synchronization procedure and the soft reset sequence described in the table below.

Table 368 Soft Reset Sequence (Normal Mode)

Host		SPC 8x6G	
Step	Host Action	Step	Action
1	<p>You can skip this step if:</p> <ul style="list-style-type: none"> the AAP_STATE in the Scratchpad 1 Register is 00 (Power-on Reset State). <p>The host checks bit [2] HOST_SOFT_RESET_RDY of the MSGU Scratchpad 2 Register via MEMBASE-I to determine it is time to issue the soft reset:</p> <ul style="list-style-type: none"> If bit [2] of the MSGU Scratchpad 2 Register is set to 1, the host goes to step 2. If bit [2] of the MSGU Scratchpad 2 Register is not set to 1, the host writes the signature 0x00001234 into the RB6 Access Register two times via MEMBASE-III, then waits for 100 milliseconds and then goes to step 2. The host does not have to check bit [2] HOST_SOFT_RESET_RDY again. <p>Bit [2] of the MSGU Scratchpad 2 Register indicates the ready status for the host to continue with a soft reset sequence. The bit indicates that the host is ready if it is set to 1. The ready status indicates that the internal CPUs of the SPC 8x6G have gone to a known/stable state, which in general will ensure that the soft reset is more reliable. In the case where the SPC 8x6G internal CPUs cannot be put to a known/stable state because of multiple faults accrued, the host can proceed to step (2).</p>		

Host		SPC 8x6G	
Step	Host Action	Step	Action
2	<p>The host disables the NMI interrupt on the IOP by writing a zero to the NMI Enable VPE0 IOP Register.</p> <p>The host disables the NMI interrupt on AAP1 by writing a zero to the NMI Enable VPE0 AAP1 Register.</p> <p>The host disables the PCIe event interrupt by writing zero to PCIe Event Interrupt Enable Register.</p> <p>The host clears the PCIe event interrupt status by first reading the PCIe Event Interrupt Register and then writing them back with the same value to clear them.</p> <p>The host disables the PCIe error interrupt by writing zero to PCIe Error Interrupt Enable Register.</p> <p>The host clears the PCIe error interrupt status by first reading the PCIe Error Interrupt Register and then writing them back with the same value to clear them.</p> <p>The host reads the MSGU Scratchpad 1 Register and saves/remembers bit [2] SFTRST_P_F setting (toggled flag indicating soft reset progress) via MEMBASE-I.</p> <p>The host writes the Soft Reset SIGNATURE (0x252ACBCD) to the Host Scratchpad 0 Register.</p>	—	—
3	<p>The host reads and sets the GSM Configuration and Reset Register as follows via the MEMBASE-III:</p> <ol style="list-style-type: none"> 1. Read the GSM Configuration and Reset Register. 2. Clear the bits (set to 0) for: <ul style="list-style-type: none"> • COM_SLV_SW_RSTB (bit 13) • QSSP_SW_RSTB (bit 12) • RAAE_SW_RSTB (bit 11) • RB_1_SW_RSTB (bit 9) • SM_SW_RSTB (bit 8) 	—	—

Host		SPC 8x6G	
Step	Host Action	Step	Action
4	<p>The host reads and saves/remembers (to be used in step 12) the GSM Read Address Parity Check Enable Register via MEMBASE-III and then clears the register (sets it to 0).</p> <p>The host reads and saves/remembers (to be used in step 12) the GSM Write Address Parity Check Enable Register via MEMBASE-III and then clears the register (sets it to 0).</p> <p>The host reads and saves/remembers (to be used in step 12) the GSM Write Data Parity Check Enable Register via MEMBASE-III and then clears the register (sets it to 0).</p>	—	—
5(a)	The host introduces a minimum delay of 10 μ s for the reset to propagate.	—	—
5(b)	<p>If the GPIO[0] signal is in driven mode during a soft reset sequence, this step is required.</p> <p>The host resets the GPIO-0 output enable to tristate through the GPIO-0 Output Control Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the GPIO-0 Output Control Register. 2. Clear the bits (set to 0) for GPIO_OUT_CTL_0[1:0]. 3. Write the register back. 	—	—
6	<p>The host resets the IOP and the AAP1 through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Clear the bits (set to 0) for PCS_IOP_SS_RSTB (bit 3) and PCS_AAP1_SS_RSTB (bit 4). 3. Write the register back. 	—	—
7	<p>The host resets the BDMA Core and the OSSP through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Clear the bits (set to 0) for BDMA_CORE_RSTB (bit 17) and OSSP_RSTB (bit 0). 3. Write the register back 	—	—
8	The host introduces a minimum delay of 10 μ s for the reset to propagate.	—	—

Host		SPC 8x6G	
Step	Host Action	Step	Action
9	<p>The host brings the BDMA and the OSSP out of reset through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Set the BDMA_CORE_RSTB (bit 17) and OSSP_RSTB (bit 0) bits to 1. 3. Write the register back. 	—	—
10	The host introduces a minimum delay of 10 μ s for the reset to propagate.	—	—
11	<p>The host reads and sets the GSM Configuration and Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the GSM Configuration and Reset Register. 2. Set the bits to 1 for: <ul style="list-style-type: none"> • COM_SLV_SW_RSTB (bit 13) • QSSP_SW_RSTB (bit 12) • RAAE_SW_RSTB (bit 11) • RB_1_SW_RSTB (bit 9) • SM_SW_RSTB (bit 8) 	—	—
12	<p>The host restores the GSM Read Address Parity Check Enable Register (saved in step 4) via MEMBASE-III .</p> <p>The host restores the GSM Write Address Parity Check Enable Register (saved in step 4) via MEMBASE-III .</p> <p>The host restores the GSM Write Data Parity Check Enable Register (saved in step 4) via MEMBASE-III .</p>	—	—
13	<p>The host brings the IOP and the AAP1 out of reset through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Set the PCS_IOP_SS_RSTB (bit 3) and the PCS_AAP1_SS_RSTB (bit 4) bits to 1. 3. Write the register back. 	—	—
14	<p>The host introduces a minimum delay of 10 μs for the IOP and AAP1 to come out of reset and the SPC 8x6G boot ROM code to start running.</p>	1	The Bootloader is running on AAP1.
		2	The Bootloader is loading the ILA from flash memory to the GSM for AAP1 and IOP.
		3	The ILA is running on AAP1. The ILA is loading AAP1 firmware from flash memory to the GSM.
		4	The ILA AAP1 releases IOP reset.
		5	The ILA is running on the IOP. Continue with the normal sequence to load firmware image and jump to firmware.

Host		SPC 8x6G	
Step	Host Action	Step	Action
15	<p>The host polls the MSGU Scratchpad 1 Register via MEMBASE-I until bit [2] SFTRST_P_F is the toggled value of host step (2) above.</p> <p>Minimum wait time is 800 msec if the boot diagnostic UART is enabled and 700 msec if the boot diagnostic UART is disabled.</p> <p>Once the MSGU Scratchpad 1 Register bit [2] SFTRST_P_F is toggled, host polls for MSGU Scratchpad 1 Register bits [1:0] AAP_STATE and Scratchpad 2 Register bits [1:0] IOP_STATE to go to 11b (Ready state).</p>	6	<p>The AAP1 ILA reads the for the Soft Reset SIGNATURE indication. If it indicates a soft reset, do not initialize the PCIe.</p> <p>For the soft reset case:</p> <ol style="list-style-type: none"> Set the MSGU Scratchpad 1 Register bits [1:0] AAP_STATE to 01b, clear bits [31:3] and toggle bit [2] SFTRST_P_F. Set the MSGU Scratchpad 2 Register bits [1:0] IOP_STATE to 01b and clear bits [31:2]. <p>For the non-Soft Reset case:</p> <ol style="list-style-type: none"> Clear (set all 32 bits [31:0] to zero) the MSGU Scratchpad 1 Register and the Scratchpad 2 Register. (Power on reset state).
		7	The AAP1 ILA continues with the normal sequence to load firmware image and jump to firmware.
—	—	—	—
16	The host clears all pending interrupts by writing all 1s to the MSGU Outbound Doorbell Clear Register (Section 10.2.4) and by writing all 0s to the MSGU Outbound Doorbell Mask Register (Section 10.2.17) via MEMBASE-I.	—	When AAP1/MSGU initialization is completed, the Scratchpad 1 Register bits [1:0] AAP_STATE to 11b (Ready state) are set.
17	The host continues with the normal SPC 8x6G Configuration Table initialization sequence as described in Section 5.2.6.1, “ Host-SPC 8x6G MPI Initialization ”.	—	When IOP MPI_APP initialization is completed, the MSGU Scratchpad 2 Register bits [1:0] IOP_STATE to 11b (Ready state) are set.

11.4.3 Soft Reset Recovery (HDA Mode)

This procedure is used for soft reset recovery if the SPC 8x6G is already operating in HDA mode or used to bring the SPC 8x6G boot ROM into the HDA mode before the host begins the HDA firmware download initialization sequence.

Table 369 Soft Reset Sequence (HDA Mode)

Host		SPC 8x6G	
Step	Host Action	Step	Action
1	<p>Skip this step if the SPC 8x6G is already in HDA mode (See Section 11.4.1) You can also skip this step if:</p> <ul style="list-style-type: none"> the AAP_STATE in the Scratchpad 1 Register is 00 (Power-on Reset State) or the Scratchpad 1 Register is 0x8002 or the AAP_STATE in the Scratchpad 1 Register is 00 (Power-on Reset State). <p>The host checks bit [2] HOST_SOFT_RESET_RDY of the MSGU Scratchpad 2 Register via MEMBASE-I to determine it is time to issue the soft reset:</p> <ul style="list-style-type: none"> If bit [2] of the MSGU Scratchpad 2 Register is set to 1, the host goes to step 2. If bit [2] of the MSGU Scratchpad 2 Register is not set to 1, the host writes the signature 0x00001234 into the RB6 Access Register two times via MEMBASE-III, then waits for 100 milliseconds and then goes to step 2. The host does not have to check bit [2] HOST_SOFT_RESET_RDY again. <p>Bit [2] of the MSGU Scratchpad 2 Register indicates the ready status for the host to continue with a soft reset sequence. The bit indicates that the host is ready if it is set to 1. The ready status indicates that the internal CPUs of the SPC 8x6G have gone to a known/stable state, which, in general, will ensure that the soft reset is more reliable. In the case where the SPC 8x6G internal CPUs cannot be put to a known/stable state because of multiple faults accrued, the host can proceed to step (2).</p>		

Host		SPC 8x6G	
Step	Host Action	Step	Action
2	<p>The host disables the NMI interrupt on the IOP by writing a zero to the NMI Enable VPE0 IOP Register.</p> <p>The host disables the NMI interrupt on AAP1 by writing a zero to the NMI Enable VPE0 AAP1 Register.</p> <p>The host disables the PCIe event interrupt by writing zero to PCIe Event Interrupt Enable Register.</p> <p>The host clears the PCIe event interrupt status by first reading the PCIe Event Interrupt Register and then writing them back with the same value to clear them.</p> <p>The host disables the PCIe error interrupt by writing zero to PCIe Error Interrupt Enable Register.</p> <p>The host clears the PCIe error interrupt status by first reading the PCIe Error Interrupt Register and then writing them back with the same value to clear them.</p> <p>The host writes the Soft Reset HDA mode SIGNATURE (0xA5AA27D7) to the Host Scratchpad 0 Register.</p>	—	—
3	<p>The host reads and sets the GSM Configuration and Reset Register as follows via MEMBASE-III :</p> <ol style="list-style-type: none"> 1. Read the GSM Configuration and Reset Register. 2. Clear the bits (set to 0) for: <ul style="list-style-type: none"> • COM_SLV_SW_RSTB (bit 13) • QSSP_SW_RSTB (bit 12) • RAAE_SW_RSTB (bit 11) • RB_1_SW_RSTB (bit 9) • SM_SW_RSTB (bit 8) 	—	—
4	<p>The host reads and saves/remembers (to be used in step 12) the GSM Read Address Parity Check Enable Register via MEMBASE-III and then clears the register (sets it to 0).</p> <p>The host reads and saves/remembers (to be used in step 12) the GSM Write Address Parity Check Enable Register via MEMBASE-III and then clears the register (sets it to 0).</p> <p>The host reads and saves/remembers (to be used in step 12) the GSM Write Data Parity Check Enable Register via MEMBASE-III and then clears the register (sets it to 0).</p>	—	—
5(a)	The host introduces a minimum delay of 10 μ s for the reset to propagate.	—	—

Host		SPC 8x6G	
Step	Host Action	Step	Action
5(b)	<p>This step is required when GPIO[0] is in driven mode during a soft reset sequence, or when entering HDA mode by an invalid ILA image as described in Section 3.21.1.</p> <p>The host resets the GPIO-0 output enable to tristate through the GPIO-0 Output Control Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the GPIO-0 Output Control Register. 2. Clear the bits (set to 0) for GPIO_OUT_CTL_0[1:0]. 3. Write the register back. 		
6	<p>The host resets the IOP and the AAP1 through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Clear the bits (set to 0) for PCS_IOP_SS_RSTB (bit 3) and PCS_AAP1_SS_RSTB (bit 4). 3. Write the register back. 	—	—
7	<p>The host resets the BDMA Core and the OSSP through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Clear the bits (set to 0) for BDMA_CORE_RSTB (bit 17) and OSSP_RSTB (bit 0). 3. Write the register back. 	—	—
8	The host introduces a minimum delay of 10 μ s for the reset to propagate.	—	—
9	<p>The host brings the BDMA and the OSSP out of reset through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Set the BDMA_CORE_RSTB (bit 17) and OSSP_RSTB (bit 0) bits to 1. 3. Write the register back. 	—	—
10	The host introduces a minimum delay of 10 μ s for the reset to propagate.	—	—

Host		SPC 8x6G	
Step	Host Action	Step	Action
11	<p>The host reads and sets the GSM Configuration and Reset Register MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the GSM Configuration and Reset Register. 2. Set the bits to 1 for: <ul style="list-style-type: none"> • COM_SLV_SW_RSTB (bit 13) • QSSP_SW_RSTB (bit 12) • RAAE_SW_RSTB (bit 11) • RB_1_SW_RSTB (bit 9) • SM_SW_RSTB (bit 8) 	—	—
12	<p>The host restores the GSM Read Address Parity Check Enable Register saved in step 4) via MEMBASE-III.</p> <p>The host restores the GSM Write Address Parity Check Enable Register (saved in step 4) via MEMBASE-III.</p> <p>The host restores the GSM Write Data Parity Check Enable Register (saved in step 4) via MEMBASE-III.</p>	—	—
13	<p>The host brings the IOP and the AAP1 out of reset through the SPC Reset Register via MEMBASE-III as follows:</p> <ol style="list-style-type: none"> 1. Read the SPC Reset Register. 2. Set the, PCS_IOP_SS_RSTB (bit 3) and the PCS_AAP1_SS_RSTB (bit 4) bits to 1. 3. Write the register back. 	—	—
14	The host introduces a minimum delay of 200 milliseconds for the IOP and AAP1 to come out of reset and the SPC 8x6G boot ROM code to start running.	1	The Bootloader is running on AAP1.
15	The host clears all pending interrupts by writing all 1s to the MSGU Outbound Doorbell Clear Register (Section 10.2.4) and by writing all 0s to the MSGU Outbound Doorbell Mask Register (Section 10.2.17) via MEMBASE-I.	—	—
16	Continue with the HDA initialization sequence described in Section 3.21.4, “ Host and SPC 8x6G Initialization Sequence ”.	—	

11.4.4 Chip Reset

A chip reset re-initializes the entire SPC 8x6G chip. The device firmware will be completely restarted. Chip reset does not preserve the SPC 8x6G's PCIe subsystem. A PCIe reset will be asserted on the PCIe bus. As the result, the host PCIe Configuration Space may be changed.

A chip reset is initiated by writing bit [31], SW_DEVICE_RSTB, of the [SPC Reset Register](#) (Section [10.3.7](#)) to zero followed by another write to set the bit to one.

The host is required to exercise new PCIe enumeration and to determine the (potentially) new PCIe Configuration Space assigned to the instance of the SPC 8x6G being reset. Once the SPC 8x6G PCIe Configuration Space is identified, the host may continue with the normal SPC 8x6G Configuration Table initialization sequence as described in Section, [5.2.6.1, “Host-SPC 8x6G MPI Initialization”](#).

11.5 SAS/SATA Error Recovery Procedures

For details about the management of SAS/SATA PHY down and SAS broadcast change events, see the following sections:

- [Section 8.2, SAS_HW_EVENT Notification](#)
- [Section 7.24, SAS_HW_EVENT_ACK Command](#)
- [Section 8.3, SSP_COMPLETION Response](#)
- [Section 8.4, SMP_COMPLETION Response](#)
- [Section 8.9, SATA_COMPLETION Response](#)
- [Section 8.10, SATA_EVENT Notification](#)
- [Section 8.11, SSP_EVENT Notification](#)

More information about SAS errors is provided in Section [11.6](#) and more information about SATA errors is provided in Section [11.7](#).

The following sections describe error reporting and the host responses associated with the PHY down, BROADCAST CHANGE, and PHY error events.

11.5.1 PHY Down Handler (External Trigger)

The PHY down handle differs depending on whether the port associated with the PHY down event is affected.

When a PHY in a narrow port is down or when the last PHY in a wide port is down, its port is directly affected. When a PHY other than the last PHY in a wide port is down, it port is not directly affected as the I/O can still resume using different PHYs.

An externally triggered PHY down is detected through a [SAS_HW_EVENT Notification](#) (see Section 8.2) in either an:

- IOP_EVENT_PHY_DOWN or
- IOP_EVENT_PORT_RECOVERY_TIMER_TMO

The IOP_EVENT_PORT_RECOVERY_TIMER_TMO event is only reported after an IOP_EVENT_PHY_DOWN and only if the Port Recovery Time is set to a non-zero value. The Port Recover Time scheme supports:

- The desired IT_NEXUS_LOSS time set for devices.
- Fault tolerance for momentary link fluctuations.

The time is defined as the maximum time between the SPC 8x6G's detection of the PHY down until the receipt of the ID_Frame (from the same remote SAS port). If the time expires before the ID_FRAME is received, the port is considered INVALID and it will not be reused. The host should notify the SPC 8x6G firmware to release the port. See Section 7.26, “[PORT_CONTROL Command](#)” for the information on setting the Port Recovery Time.

The **IOP_EVENT_PORT_RECOVERY_TIMER_TMO** event is reported following the IOP_EVENT_PHY_DOWN (PORT_LOSTCOMM) event when the PHY/port does not recover after Port Recovery Time.

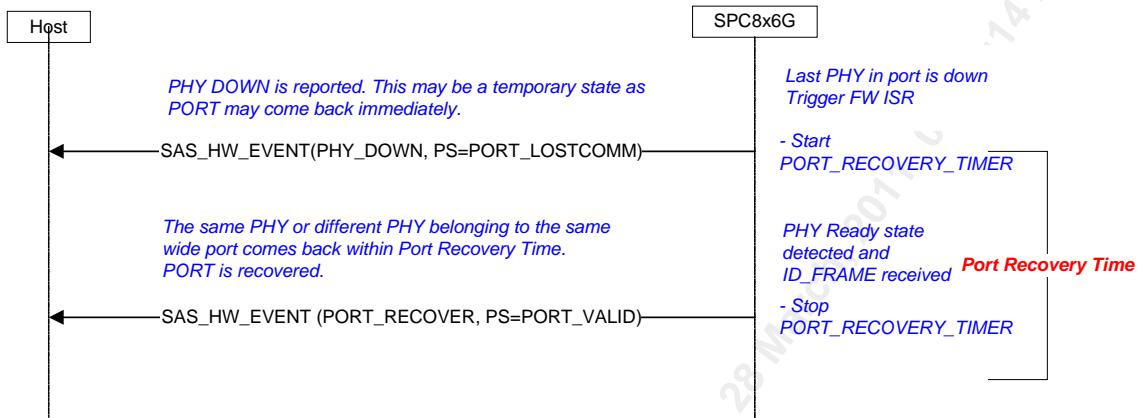
Included in the [SAS_HW_EVENT Notification](#) are the fields:

- PHYID
- PORT_ID
- NPIP (Number of PHYS in the port)
- PS (Port state)

11.5.1.1 Last PHY Down in A Port with Port Recovery Time Enabled

The following diagram shows the flow for a PHY down of the last PHY in port event that is recovered since the PHY up and the ID_Frame is received within Port Recovery Time.

Figure 51 PHY Down (of the Last PHY in a Port) That Recovered

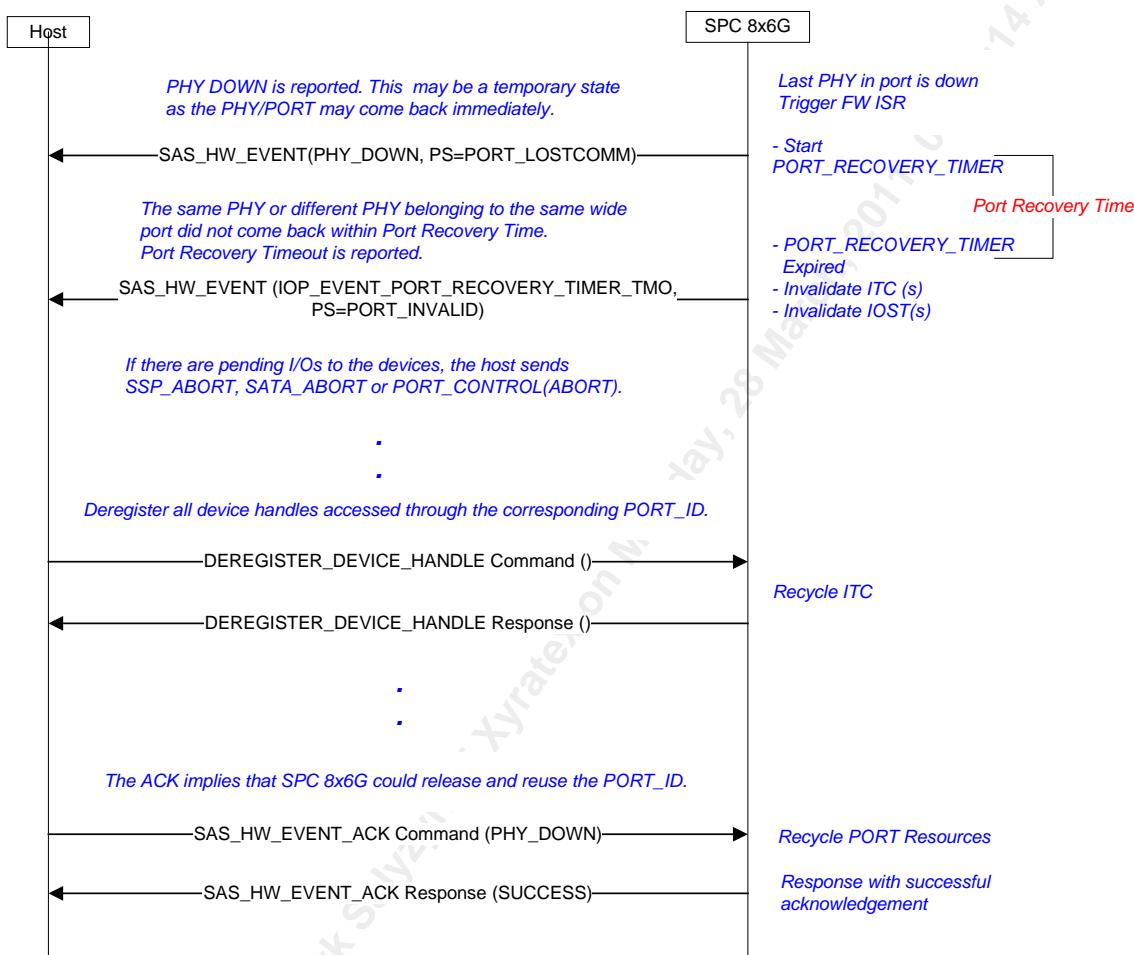


As shown in Figure 51, if the PHY ready state is re-established and the ID_Frame is received, the SPC 8x6G will notify host with an IOP_EVENT_PORT_RECOVER event. Between the IOP_EVENT_PHY_DOWN and the IOP_EVENT_PORT_RECOVER events, some I/O operations may fail. These are reported in either the [SSP_EVENT Notification](#) (Section 8.11), [SSP_COMPLETION Response](#) (8.3), [SATA_EVENT Notification](#) (Section 8.10), or [SATA_COMPLETION Response](#) (Section 8.9).

In some cases where the PHY/port is recovered fast enough (less than the Port Recovery Time and less than the setting of IT Nexus Timeout with the device) the I/O may proceed successfully.

The following diagram shows the PHY down of a last PHY in a port that is not recovered within the Port Recovery Time.

Figure 52 PHY Down Sequence with No Recovery After the Port Recovery Time



As shown in Figure 52, following the initial IOP_EVENT_PHY_DOWN event with the PS (Port State) field set to PORT_LOSTCOMM, the PHY/port has not recovered within the Port Recovery Time. As a result, the IOP_EVENT_PORT_RECOVERY_TIMER_TMO event is reported with a PS field set to PORT_INVALID.

Upon receiving an IOP_EVENT_PHY_DOWN (PORT_LOSTCOMM), at a minimum, the host should stop issuing new I/Os to devices accessed through the port in question.

Upon receiving an IOP_EVENT_PORT_RECOVERY_TIMER_TMO and before acknowledging the event with a [SAS_HW_EVENT_ACK Command](#) (Section 7.24), which will release the PORT_ID, the host must perform some clean-up, including:

- Abort pending I/Os in devices (DEVICE_IDS) associated with that PORT_ID if there are still pending I/Os that have not been reported with errors.

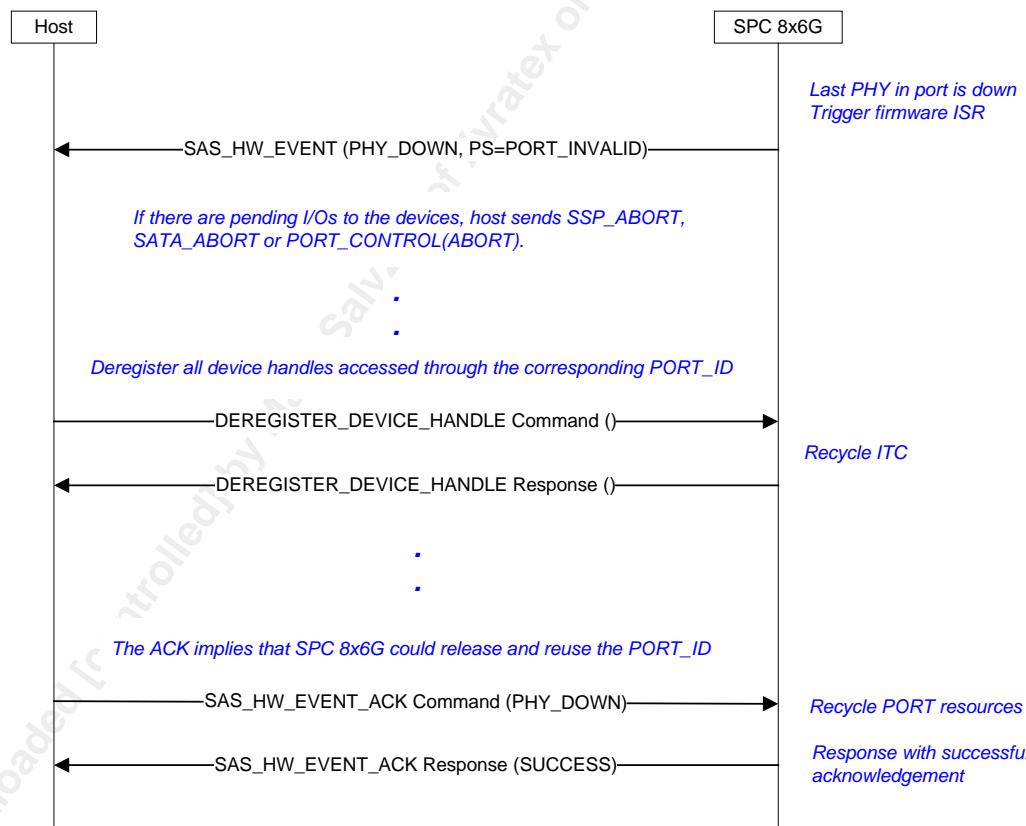
- Deregister the all DEVICE_IDS associated with the PORT_ID.

The SPC 8x6G will not report the IOP_EVENT_SAS_PHY_UP event in the **SAS_HW_EVENT Notification** (Section 8.2) of any of the PHYs with the same PORT_ID that have not been released through a **SAS_HW_EVENT_ACK Command**. The IOP_EVENT_SAS_PHY_UP event will be reported immediately if an unused PORT_ID is available. When an IOP_EVENT_SAS_PHY_UP event of any of these PHYs is eventually reported to the host through a **SAS_HW_EVENT Notification**, the PORT_ID is not guaranteed to be persistent. The SPC 8x6G guarantees that at any moment, on the host side, there will not be more than one PORT_ID with the PORT_VALID state assigned to a particular PHY. That is, a PHY cannot belong to more than one PORT_ID with a PORT_VALID state.

11.5.1.2 Last PHY Down in a Port with Port Recovery Time Set to Zero

The following diagram shows the PHY down of the last PHY in a port with the Port Recovery Time set to zero.

Figure 53 Last PHY Down in a Port with Port Recovery Time Set to Zero



As shown in [Figure 53](#), upon receiving an IOP_EVENT_PHY_DOWN (PORT_INVALID), at a minimum, the host should stop issuing new I/Os to devices accessed through the port in question.

Before acknowledging the event with a [SAS_HW_EVENT_ACK Command](#) (Section [7.24](#)), which will release the PORT_ID, the host must perform some clean-up, including:

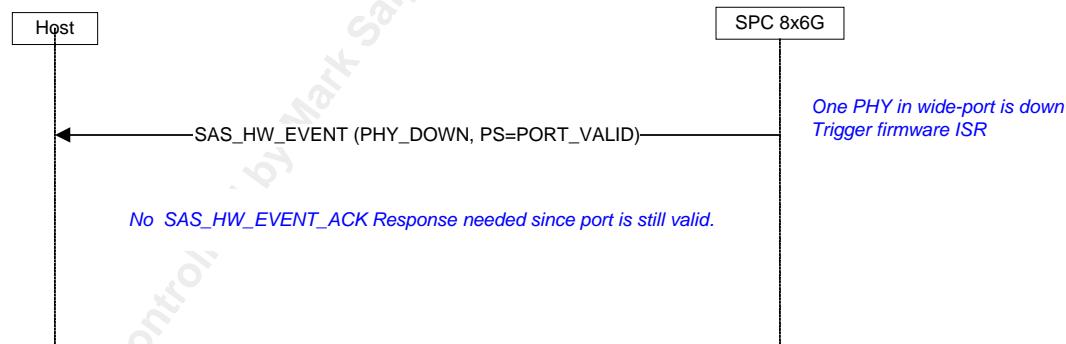
- Abort all pending I/Os in the devices (DEVICE_IDS) associated with the PORT_ID if there are still pending I/Os that have not been reported with errors.
- Deregister all DEVICE_IDS associated with the PORT_ID.

The SPC 8x6G will not report the IOP_EVENT_SAS_PHY_UP event in the [SAS_HW_EVENT Notification](#) (Section [8.2](#)) of any of the PHYs with the same PORT_ID that has not been released through a [SAS_HW_EVENT_ACK Command](#). The IOP_EVENT_SAS_PHY_UP event will be reported immediately if an unused PORT_ID is available. When the IOP_EVENT_SAS_PHY_UP event of any of these PHYs is eventually reported to the host through a [SAS_HW_EVENT Notification](#), the PORT_ID is not guaranteed to be persistent. The SPC 8x6G guarantees that at any moment, on the host side, there will not be more than one PORT_ID with the PORT_VALID state assigned to a particular PHY, that is, a PHY cannot belong to more than one PORT_ID with a PORT_VALID state.

11.5.1.3 PHY Down of a PHY in a Wide Port

The following diagram shows a PHY down of a PHY other than the last PHY of a wide port where some other PHYs in the port are still operational.

Figure 54 PHY Down Sequence of One of the PHYs in a Wide Port



As shown in [Figure 54](#) above, the host **does not** need to send an acknowledgement via a [SAS_HW_EVENT_ACK Command](#) (as described in Section [7.24](#)) for the PHY down event since the port is still valid, that is, some other PHYs are still up for the port.

11.5.2 Local PHY Control Link/Hard Reset Handler (Host Initiated)

This section describes the SAS LINK RESET, SAS HARD RESET and SATA LINK RESET local PHY control sequence and handler using the [LOCAL_PHY_CONTROL Command](#) described in Section [7.18](#).

A scheme based on port reset time is used to avoid the unnecessary disruption to the port (PS is becoming PORT_INVALID) when all local PHYs in a port are reset in a sequence. That is, subsequent PHYs in the port are reset without waiting for the PHY up of the previous PHY to be received.

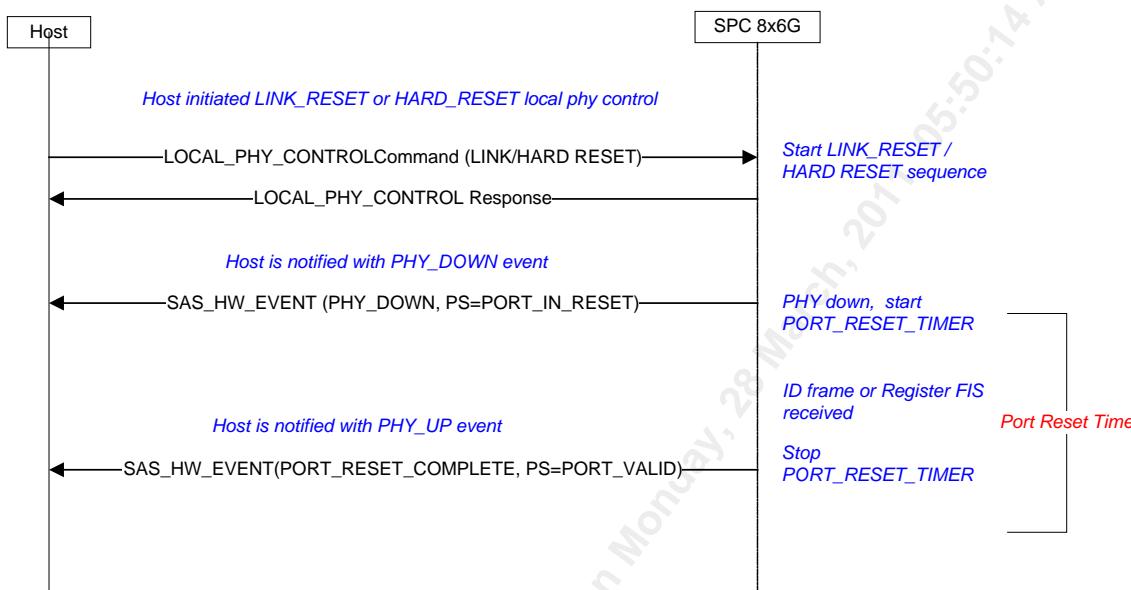
The SPC 8x6G implements the port reset timer on a per-port basis. The port reset timer is started when the last PHY in the port is down and cancelled when a PHY up is detected in any of the PHYs in that port.

The port reset time requirement is different between SAS and SATA. The port reset time value can be changed using the [PORT_CONTROL Command](#) described in Section [7.26](#). For SAS, the port reset time should be set at least 300 milliseconds to satisfy the SAS requirement of a maximum of 250 milliseconds for the remote node to start the PHY reset sequence. (See the T10/1760-D, SAS-2 draft, Revision 14b, Section 4.7.1). The SPC 8x6G default value for SAS port reset time is 300 milliseconds. For SATA, the port reset time should be set at least TBD to satisfy the ATA drive requirement (reference TBD). The SPC 8x6G default value for SATA port reset time is 2*1000 milliseconds (2 seconds). There is no option to disable the port reset time scheme. In other words, the port reset time should not be set to zero otherwise host-initiated local PHY control for a link/hard reset will always cause the port to go into a PORT_INVALID state.

11.5.2.1 Narrow Port Successful Link/Hard Reset

The following diagram shows the flow of a successful local PHY control link/hard reset initiated by the host to a PHY in a narrow port.

Figure 55 Narrow Port Successful Local PHY Control Link/Hard Reset



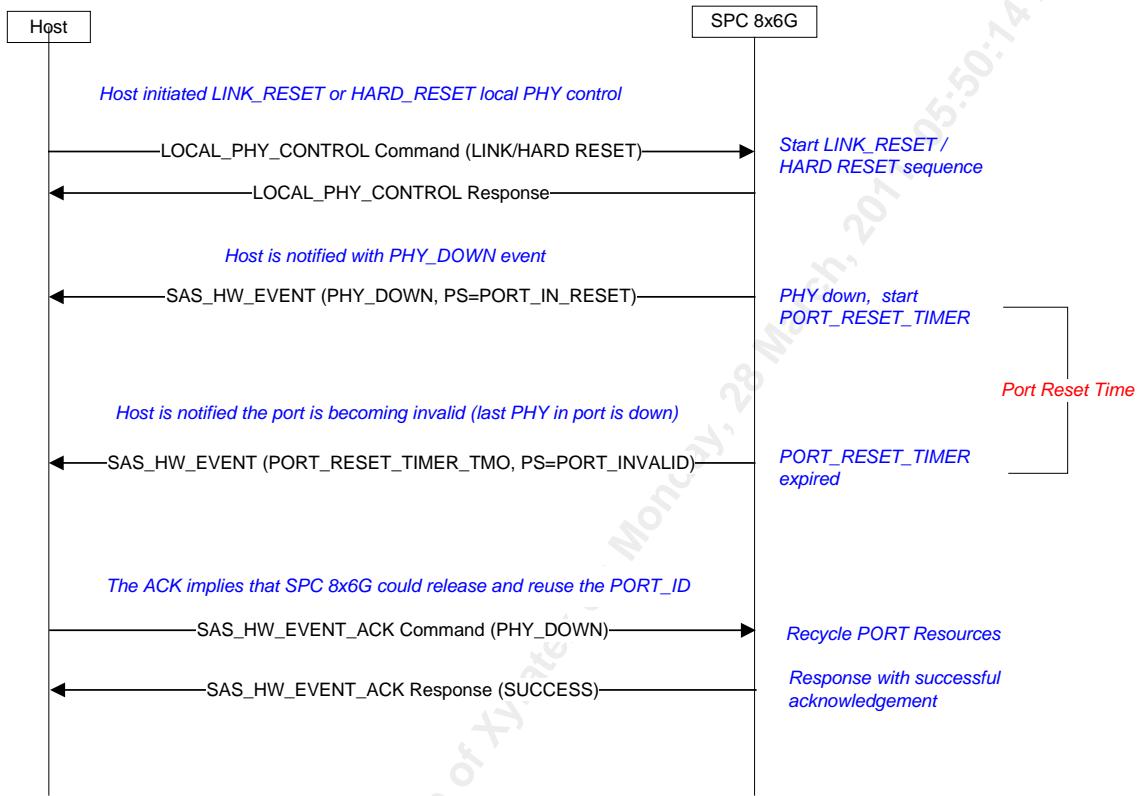
As shown in [Figure 55](#), the SPC 8x6G issues the link/hard reset to the local PHY. When a PHY down is detected, the SPC 8x6G starts the port reset timer and notifies the host with an IOP_EVENT_PHY_DOWN event in the [SAS_HW_EVENT Notification](#) (Section 8.2), with the PS field set to PORT_IN_RESET. During this time, new I/Os issued to the SPC 8x6G will be returned with the error status, IO_PORT_IN_RESET, in the [SSP_COMPLETION Response](#) (8.3), [SATA_COMPLETION Response](#) (Section 8.9), or the [SMP_COMPLETION Response](#) (Section 8.4).

When the identify address frame of the directly-attached SAS device is detected or the Register FIS of the directly-attached SATA device is detected, the SPC 8x6G stops the port reset timer and notifies the host with an IO_EVENT_PORT_RESET_COMPLETE event in the [SAS_HW_EVENT Notification](#), with PS set to PORT_VALID. At this time, I/Os can resume using this port.

11.5.2.2 Narrow Port Unsuccessful Link/Hard Reset

[Figure 56](#) shows the flow of an unsuccessful local PHY control link/hard reset initiated by the host to a PHY in a narrow port.

Figure 56 Narrow Port Unsuccessful Local PHY Control Link/Hard Reset



As shown in Figure 56, the SPC 8x6G issues the link/hard reset to the local PHY. When a PHY down is detected, the SPC 8x6G starts the port reset timer and notifies the host with an IOP_EVENT_PHY_DOWN event in the **SAS_HW_EVENT Notification** (Section 8.2), with PS set to PORT_IN_RESET. During this time, new I/Os issued to the SPC 8x6G will be returned with the IO_PORT_IN_RESET error status in the **SSP_COMPLETION Response** (8.3), **SATA_COMPLETION Response** (Section 8.9), or the **SMP_COMPLETION Response** (Section 8.4).

When the identify address frame of the directly-attached SAS device or the Register FIS of the directly-attached SATA device is not detected within the port reset time, the SPC 8x6G notifies the host with a PORT_RESET_TIMER_TMO event in the **SAS_HW_EVENT Notification**, with PS set to PORT_INVALID.

Before acknowledging the event with a **SAS_HW_EVENT_ACK Command** (Section 7.24), which will release the PORT_ID, the host must perform some clean-up, including:

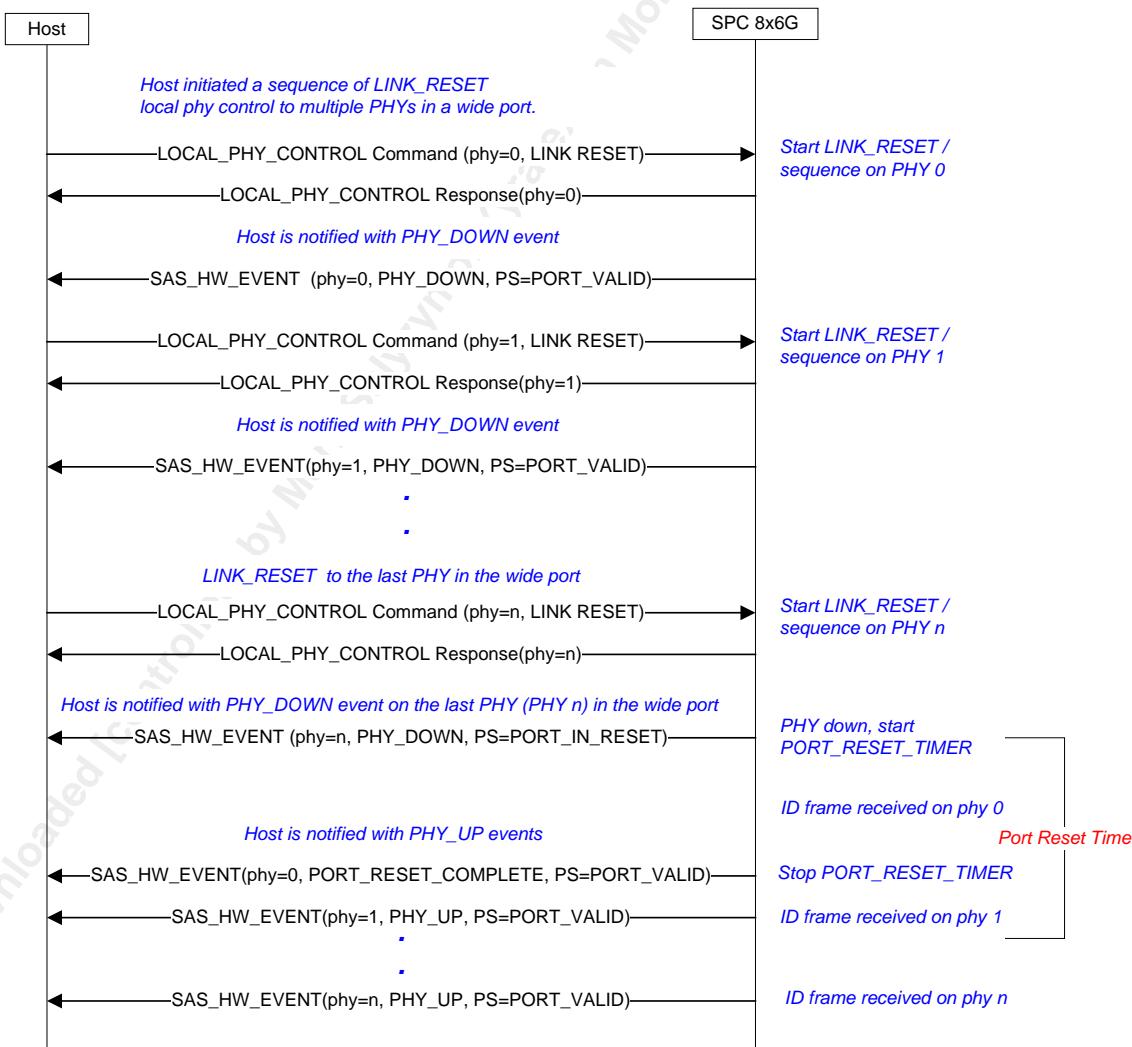
- Abort all pending I/Os in the devices (DEVICE_IDS) associated with the PORT_ID if there are still pending I/Os that have not been reported with errors.
- Deregister all DEVICE_IDS associated with the PORT_ID.

The SPC 8x6G will not report the IOP_EVENT_SAS_PHY_UP event in the **SAS_HW_EVENT Notification** of any of the PHYs with the same PORT_ID that have not been released though a **SAS_HW_EVENT_ACK Command**. The IOP_EVENT_SAS_PHY_UP event will be reported immediately if an unused PORT_IDs are available. When the IOP_EVENT_SAS_PHY_UP event of any of these PHYs is eventually reported to the host through a **SAS_HW_EVENT Notification**, the PORT_ID is not guaranteed to be persistent. The SPC 8x6G guarantees that at any moment, on the host side, there will not be more than one PORT_ID with the PORT_VALID state assigned to a particular PHY. In other words, a PHY cannot belong to more than one PORT_ID with a PORT_VALID state.

11.5.2.3 Wide Port Successful Link Reset

Figure 57 shows the flow of a successful local PHY control link reset initiated by the host in sequence to all PHYs in a wide port.

Figure 57 Wide Port Successful Local PHY Control Link Reset



As shown in [Figure 57](#), the SPC 8x6G issues a link reset to the local PHYs. When a PHY down is detected on the last PHY (PHY-*n* in the above figure), the SPC 8x6G starts the port reset timer and notifies the host with an IOP_EVENT_PHY_DOWN event in the [SAS_HW_EVENT Notification](#) (Section 8.2), with PS is set to PORT_IN_RESET. During this time, new I/Os issued to the SPC 8x6G will be returned with an IO_PORT_IN_RESET error status in the [SSP_COMPLETION Response](#) (8.3), [SATA_COMPLETION Response](#) (Section 8.9), or the [SMP_COMPLETION Response](#) (Section 8.4).

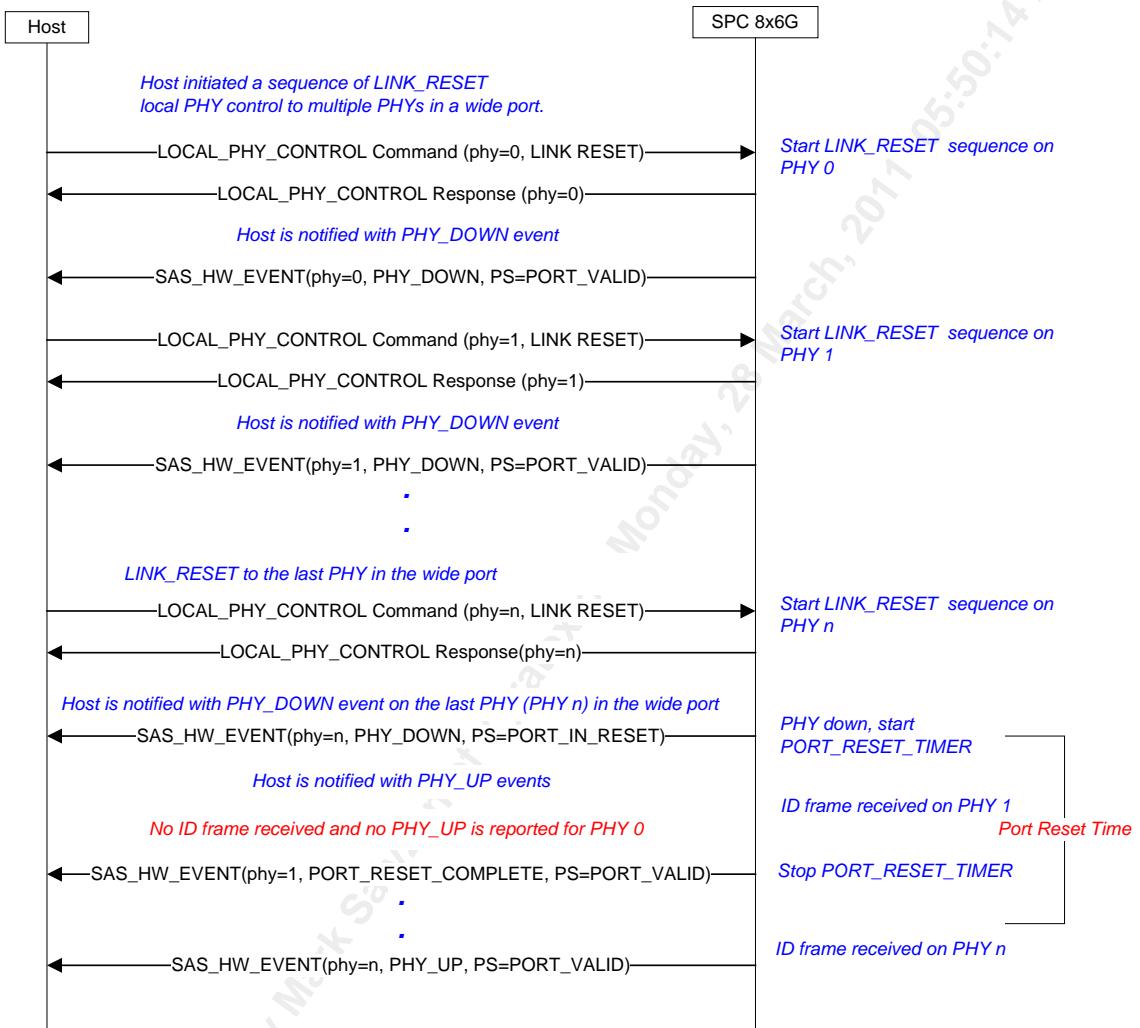
When the identify address frame of the directly-attached SAS device is detected on any of the PHYs in the wide port (on a different PHY, PHY-0 in the above figure), the SPC 8x6G stops the port reset timer and notifies the host with an IO_EVENT_PORT_RESET_COMPLETE event in the [SAS_HW_EVENT Notification](#), with PS is set to PORT_VALID. At this time, I/Os can resume using this port.

On the subsequent receipt of an identify address frame on another PHY belonging to the same wide port, the SPC 8x6G will notify the host with an IOP_EVENT_SAS_PHY_UP event in the [SAS_HW_EVENT Notification](#), with PS is set to PORT_VALID.

11.5.2.4 Wide Port Partial Successful Link Reset

The following figure shows the flow of a partial successful (no PHY up on one of the PHYs) local PHY control link reset initiated by the host in sequence to all PHYs in a wide port.

Figure 58 Wide Port Partial Successful Local PHY Control Link Reset



As shown in Figure 58, the SPC 8x6G issues the link reset to the local PHYs. When a PHY down is detected on the last PHY (PHY-n in the above figure), the SPC 8x6G starts the port reset timer and notifies the host with an IOP_EVENT_PHY_DOWN event in the **SAS_HW_EVENT Notification** (Section 8.2), with PS set to PORT_IN_RESET. During this time, I/Os issued to the SPC 8x6G will be returned with the IO_PORT_IN_RESET error status in the **SSP_COMPLETION Response** (8.3), **SATA_COMPLETION Response** (Section 8.9), or the **SMP_COMPLETION Response** (Section 8.4).

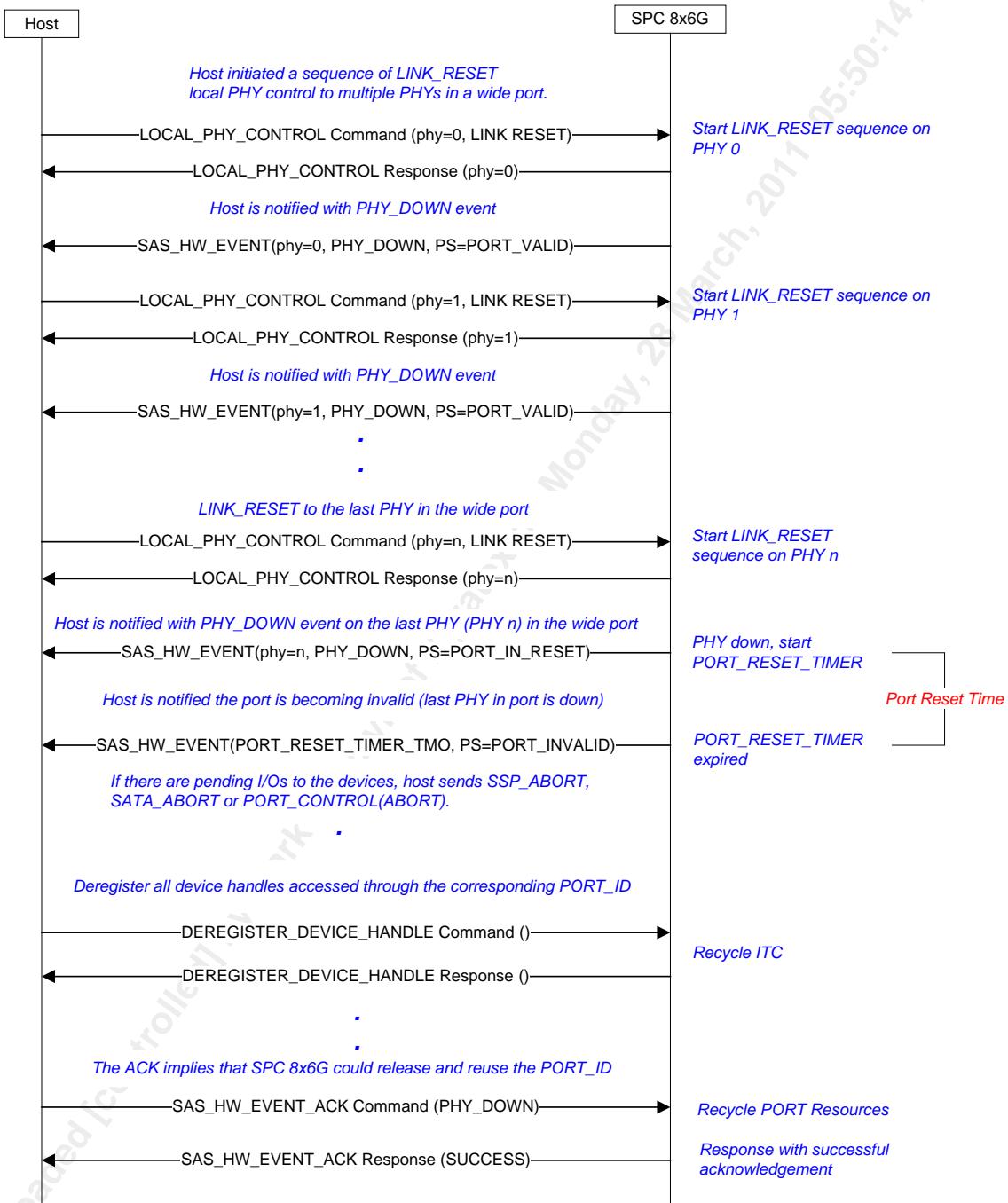
When the identify address frame of the directly-attached SAS device is detected on any of the PHYs in the wide port (on a different PHY, PHY-1 in the above figure), the SPC 8x6G stops the port reset timer and notifies the host with the IO_EVENT_PORT_RESET_COMPLETE event in the [SAS_HW_EVENT Notification](#), with PS is set to PORT_VALID. Please note that no identify address frame is detected at PHY-0. At this time, I/Os can resume using this port.

On the subsequent receipt of an identify address frame on the other PHY belonging to the same wide port, the SPC 8x6G will notify the host with an IOP_EVENT_SAS_PHY_UP event in the [SAS_HW_EVENT Notification](#), with PS is set to PORT_VALID.

11.5.2.5 Wide Port Unsuccessful Link Reset

[Figure 59](#) shows the flow of an unsuccessful (no PHY up on any of the PHYs) local PHY control link reset initiated by the host in sequence with all PHYs in a wide port.

Figure 59 Wide Port Unsuccessful Local PHY Control Link Reset



As shown in [Figure 59](#), the SPC 8x6G issues the link reset to the local PHYs. When a PHY down is detected on the last PHY (PHY-*n* in the above diagram), the SPC 8x6G starts the port reset timer and notifies the host with an IOP_EVENT_PHY_DOWN event in the [SAS_HW_EVENT Notification](#) (Section 8.2), with PS is set to PORT_IN_RESET. During this time, I/Os issued to the SPC 8x6G will be returned with the IO_PORT_IN_RESET error status in the [SSP_COMPLETION Response](#) (8.3), [SATA_COMPLETION Response](#) (Section 8.9), or the [SMP_COMPLETION Response](#) (Section 8.4).

When the identify address frame of the directly-attached SAS device is not detected within the port reset time, the SPC 8x6G notifies the host with the PORT_REST_TIMER_TMO event in the [SAS_HW_EVENT Notification](#), with PS is set to PORT_INVALID.

Before acknowledging the event with a [SAS_HW_EVENT_ACK Command](#) (Section 7.24), which will release the PORT_ID, the host must perform some clean-up, including:

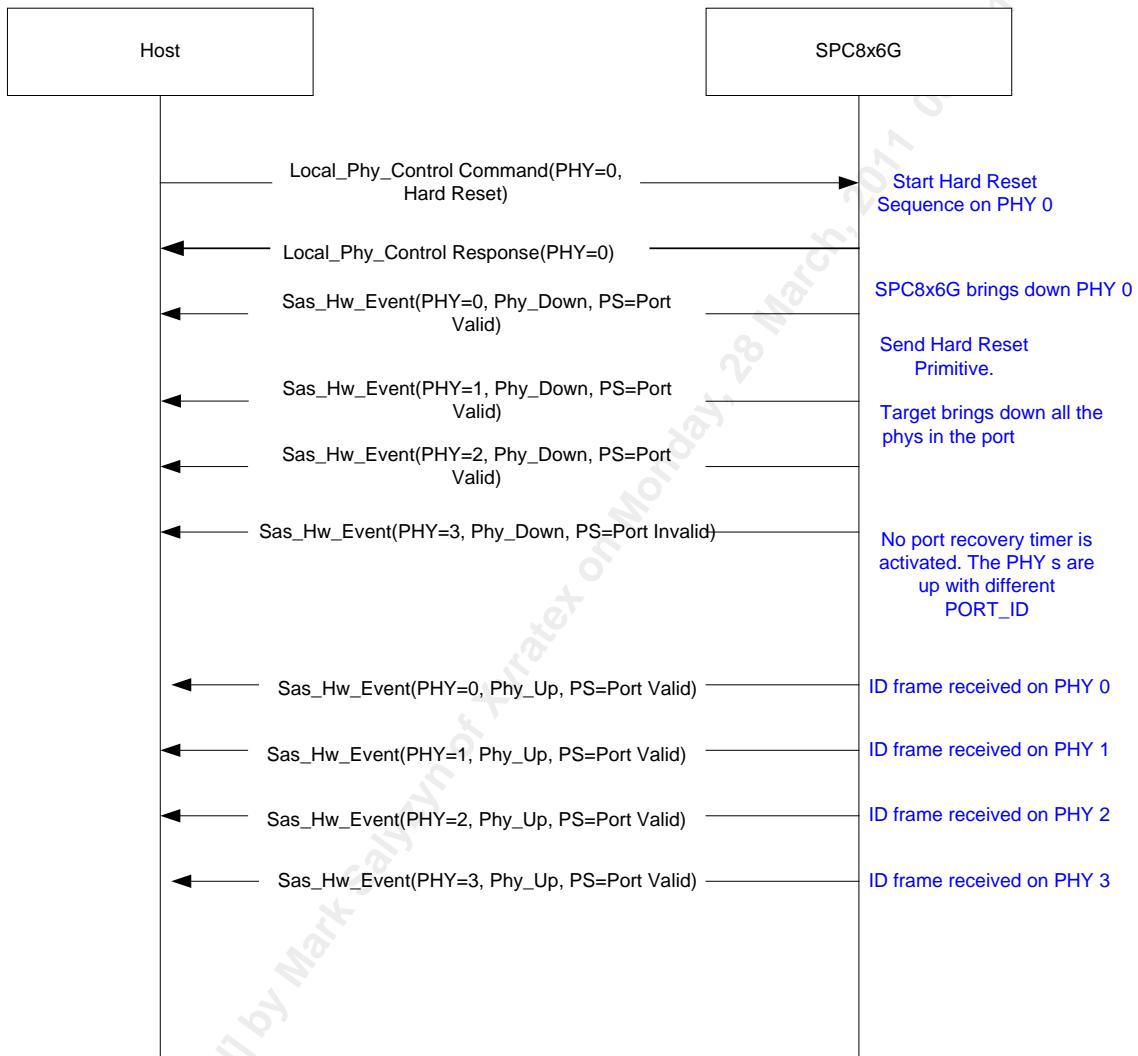
- Abort all pending I/Os in the devices (DEVICE_IDS) associated with the PORT_ID if there are still pending I/Os that have not been reported with errors.
- Deregister all DEVICE_IDS associated with the PORT_ID.

The SPC 8x6G will not report the IOP_EVENT_SAS_PHY_UP event in the [SAS_HW_EVENT Notification](#) of any of the PHYs with the same PORT_ID that have not been released though a [SAS_HW_EVENT_ACK Command](#). The IOP_EVENT_SAS_PHY_UP event will be reported immediately if an unused PORT_ID is available. When the IOP_EVENT_SAS_PHY_UP event of any of these PHYs is eventually reported to the host through a [SAS_HW_EVENT Notification](#), the PORT_ID is not guaranteed to be persistent. The SPC 8x6G guarantees that at any moment, on the host side, there will not be more than one PORT_ID with the PORT_VALID state assigned to a particular PHY. That is, a PHY cannot belong to more than one PORT_ID with a PORT_VALID state.

11.5.2.6 Wide Port Successful Hard Reset

[Figure 60](#) shows the flow of a successful local PHY control hard reset initiated by the host in sequence with all PHYs in a wide port without port recovery timer.

Figure 60 Wide Port Successful Local PHY Control Hard Reset Without Port Recovery Timer



As shown in Figure 60, the SPC 8x6G issues a hard reset to the local PHY. Then, the SPC8x6G brings down PHY0 and sends a hard reset primitive. The target brings down all the PHYs in that port. When a PHY down is detected on the last PHY (PHY-3 in the above figure), the SPC 8x6G notifies the host with an IOP_EVENT_PHY_DOWN event in the [SAS_HW_EVENT Notification](#) (Section 8.2), with PS set to PORT_INVALID. The new I/Os issued after the last IOP_EVENT_PHY_DOWN to the SPC 8x6G will be returned with an IO_DS_NON_OPERATIONAL error status in the [SSP_COMPLETION Response](#) (Section 8.3), [SATA_COMPLETION Response](#) (Section 8.9), or the [SMP_COMPLETION Response](#) (Section 8.4). When the identify address frame of the directly-attached SAS device is detected on any of the PHYs in the wide port (on a different PHY, PHY-0 in the above figure), the SPC 8x6G s notifies the host with a PHY_UP event in the [SAS_HW_EVENT Notification](#), with PS set to PORT_VALID. At this time I/Os can resume using this port.

On the subsequent receipt of an identify address frame on another PHY belonging to the same wide port, the SPC 8x6G will notify the host with an PHY_UP event in the **SAS_HW_EVENT Notification**, with PS set to PORT_VALID.

Figure 61 shows the flow of a successful local PHY control hard reset initiated by the host in sequence with all PHYs in a wide port with the port recovery timer.

Figure 61 Wide Port Successful Local PHY Control Hard Reset With Port Recovery Timer

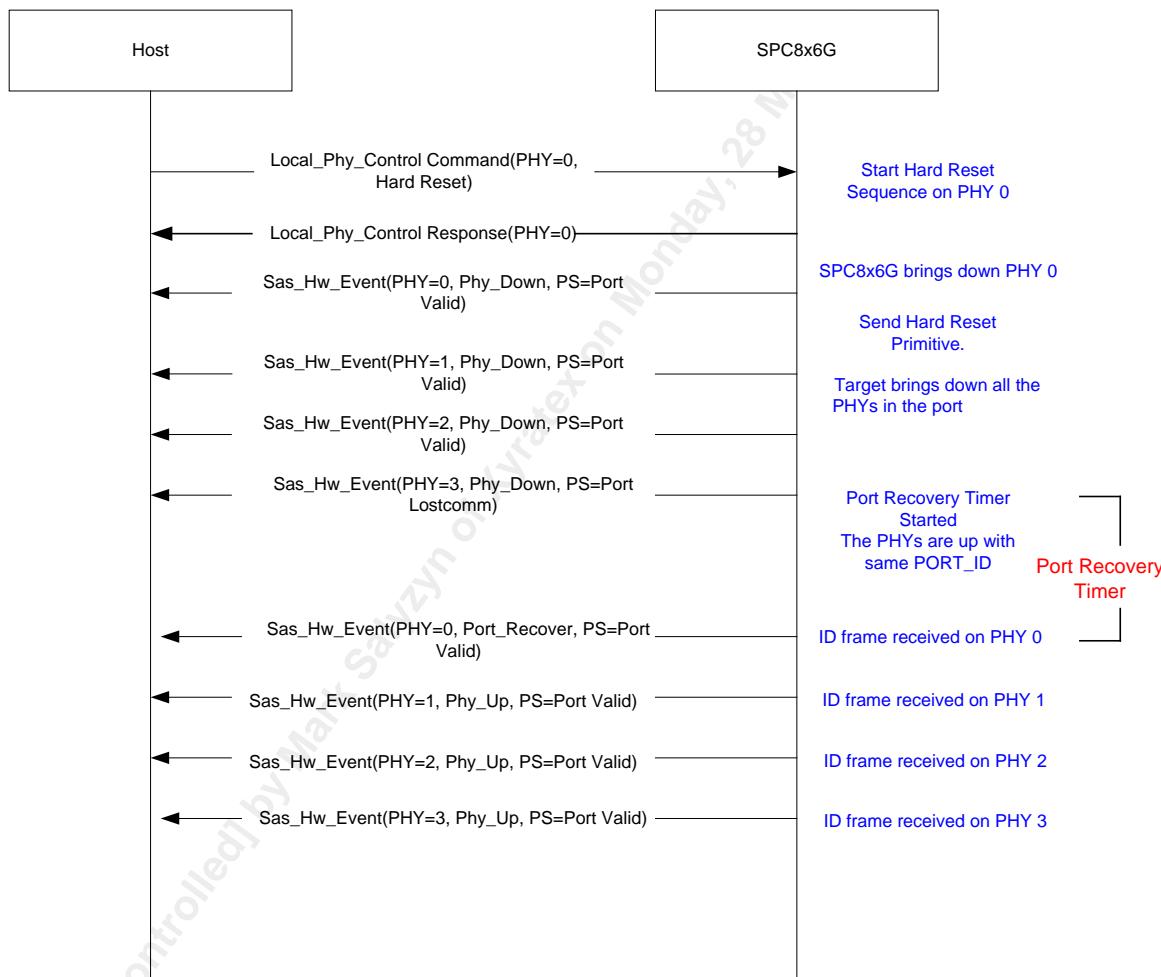


Figure 61 shows the flow of a successful local PHY control hard reset initiated by the host in a wide port.

The SPC 8x6G issues a hard reset to the local PHY. Then the SPC8x6G brings down PHY0 and sends a hard reset primitive. The target brings down all the PHYs in that port. When a PHY down is detected on the last PHY (PHY-3 in the above figure), the SPC 8x6G notifies the host with an IOP_EVENT_PHY_DOWN event in the [SAS_HW_EVENT Notification](#) (Section 8.2), with PS set to PORT_LOSTCOMM. The SPC 8x6G starts the port recovery timer. The host is advised not to issue any new I/Os after the last IOP_EVENT_PHY_DOWN to the SPC 8x6G. See [Figure 51](#) for more details.

When the identify address frame of the directly-attached SAS device is detected on any of the PHYs in the wide port (on a different PHY, PHY-0 in the above figure), the SPC 8x6G stops the port recovery timer and notifies the host with an IOP_EVENT_PORT_RECOVER and PHY_UP event in the [SAS_HW_EVENT Notification](#), with PS set to PORT_VALID. At this time I/Os can resume using this port.

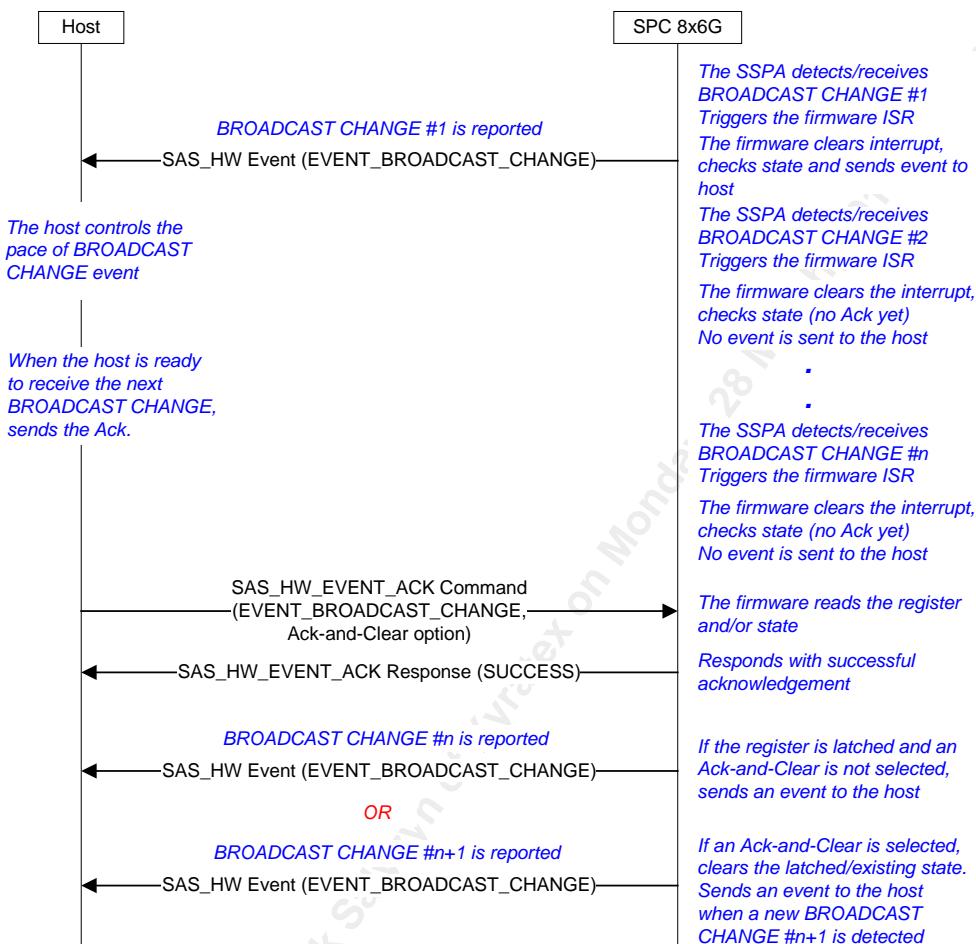
On the subsequent receipt of an identify address frame on another PHY belonging to the same wide port, the SPC 8x6G will notify the host with a PHY_UP event in the [SAS_HW_EVENT Notification](#), with PS set to PORT_VALID.

11.5.3 BROADCAST CHANGE Handler

The first detection of a BROADCAST_CHANGE by the SPC 8x6G will result in an IOP_EVENT_BROADCAST_CHANGE event generated for the host through a [SAS_HW_EVENT Notification](#) of (Section 8.2).

An event and acknowledge handshake between the SPC 8x6G and the host is used to allow the host to pace the rate that the event messages are generated to the host. The host sends a [SAS_HW_EVENT_ACK Command](#) (Section 7.24) to acknowledge the receipt of a BROADCAST_CHANGE and passes the SPC 8x6G option for how to notify the host on the next BROADCAST_CHANGE detection.

Figure 62 BROADCAST CHANGE Acknowledgement Sequence



This acknowledgement flow and synchronization guarantees that the first and last BROADCAST CHANGE events received on this PHY will always be available for the host to receive. When sending a **SAS_HW_EVENT_ACK Command**, the host has the option to do an Ack-and-Clear operation where the host directs the SPC 8x6G to ignore all previously-latched BROADCAST_CHANGE events, clear all previous states, and only report new occurrences of BROADCAST_CHANGE events.

Typically, the host will need to do some sort of rediscovery when a BROADCAST CHANGE event is detected on the PHY. Details of discovery process are described in the T10 SAS specification.

11.5.4 PHY Error Handler

This section describes the error notification related to the following PHY error (IOP_EVENT_PHY_ERR_XX) events:

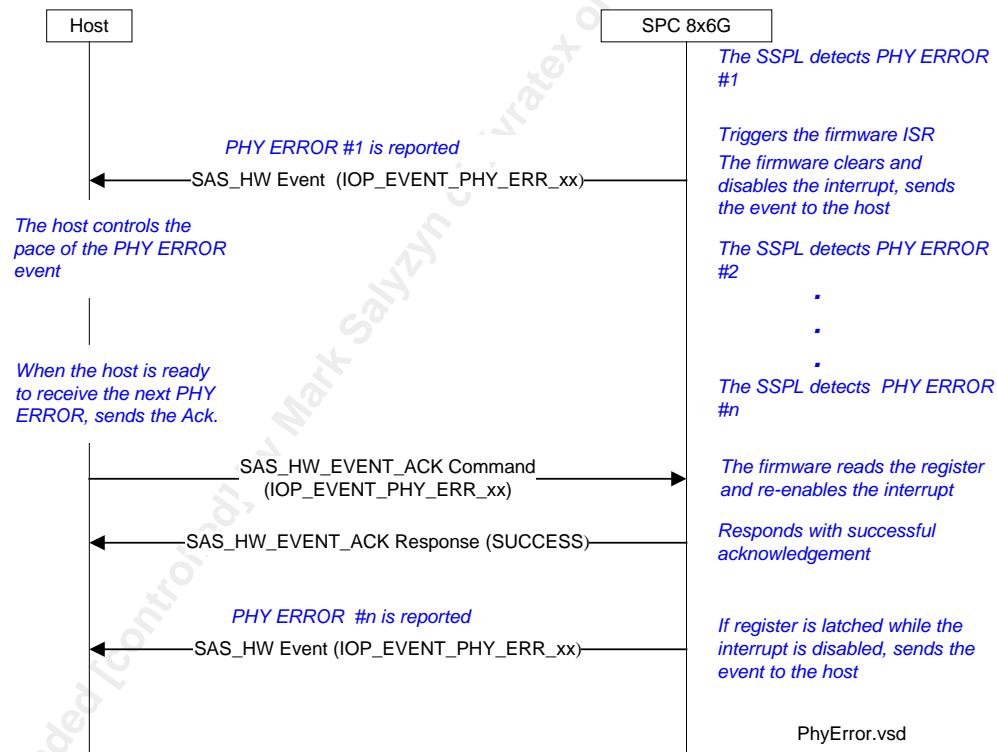
- IOP_EVENT_PHY_ERR_INVALID_DWORD

- IOP_EVENT_PHY_ERR_DISPARITY_ERROR.
- IOP_EVENT_PHY_ERR_CODE_VIOLATION
- IOP_EVENT_PHY_ERR_LOSS_OF_DWORD_SYNCH
- IOP_EVENT_PHY_ERR_PHY_RESET_FAILED
- IOP_EVENT_INBOUND_CRC_ERROR

The first detection of an IOP_EVENT_PHY_ERR_XX event by the SPC 8x6G will result an IOP_EVENT_PHY_ERR_XX event generated for the host through a [SAS_HW_EVENT Notification](#) (Section 8.2).

An event and acknowledge handshake between the SPC 8x6G and the host is used to allow the host to pace the rate that the event messages are generated to the host. The host sends a [SAS_HW_EVENT_ACK Command](#) (Section 7.24) to acknowledge the reception of the IOP_EVENT_PHY_ERR_XX event.

Figure 63 PHY Error Acknowledgement Sequence



The PHY error counts can also be accessed by the host using the SPC 8x6G PCIe registers. See Sections 10.6.5 to 10.6.10.

The host may decide to disable the PHY by sending a PHY_STOP Command if excessive PHY error is detected on a particular PHY.

11.5.5 Hard Reset Received Handler

This section describes the SPC 8x6G's handling of a Hard Reset Received SAS primitive in both target and initiator modes.

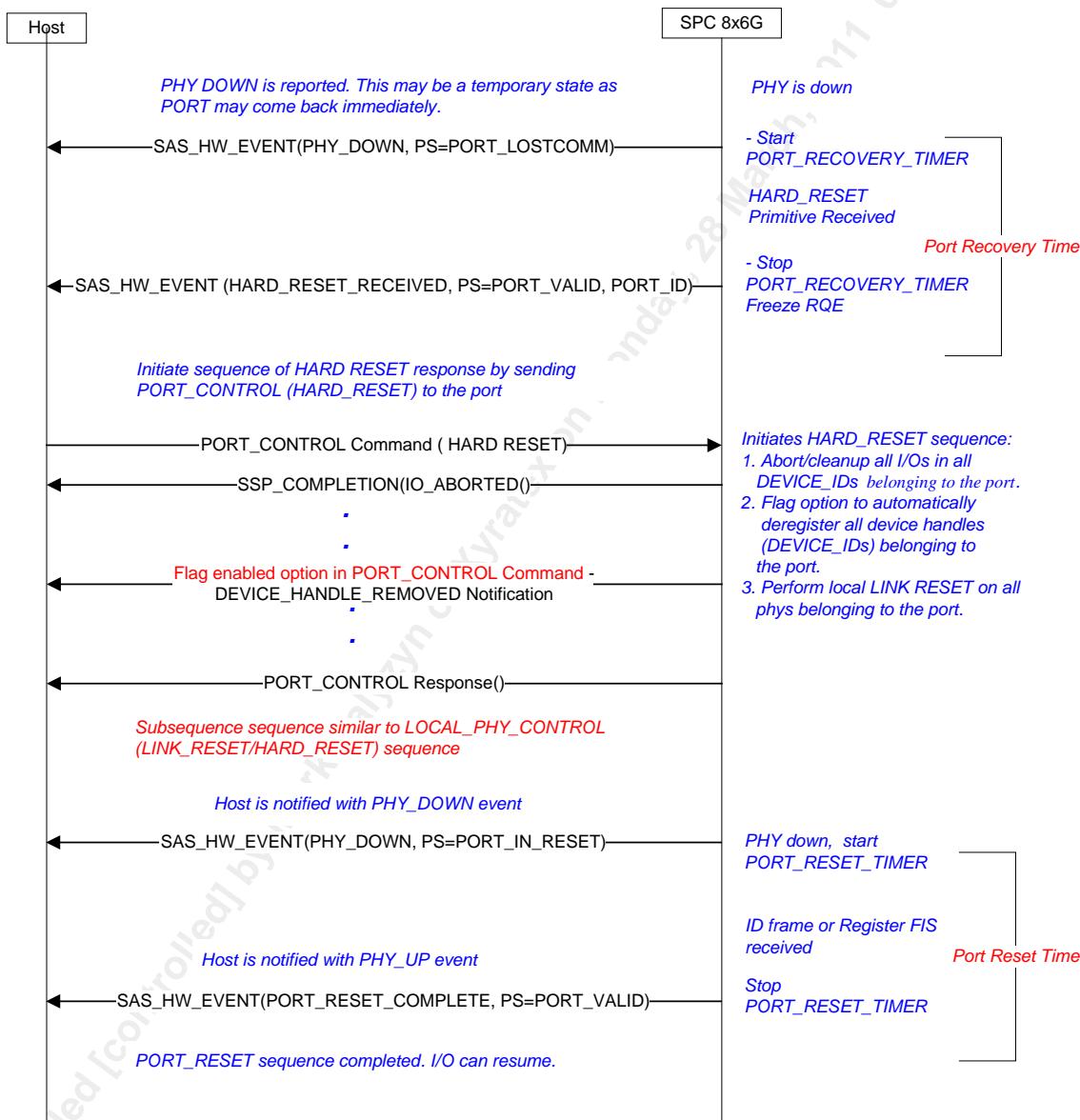
The following subsections describe the sequence for:

- Narrow port with port recovery time enabled
- Narrow port with port recovery time disabled
- Wide port with port recovery time enabled or disabled

11.5.5.1 Narrow Port Hard Reset Received - Port Recovery Time Enabled

Figure 64 shows the handling of a narrow port Hard Reset with the port recovery time enabled.

Figure 64 Narrow Port– Hard Reset Received – Port Recovery Time Enabled

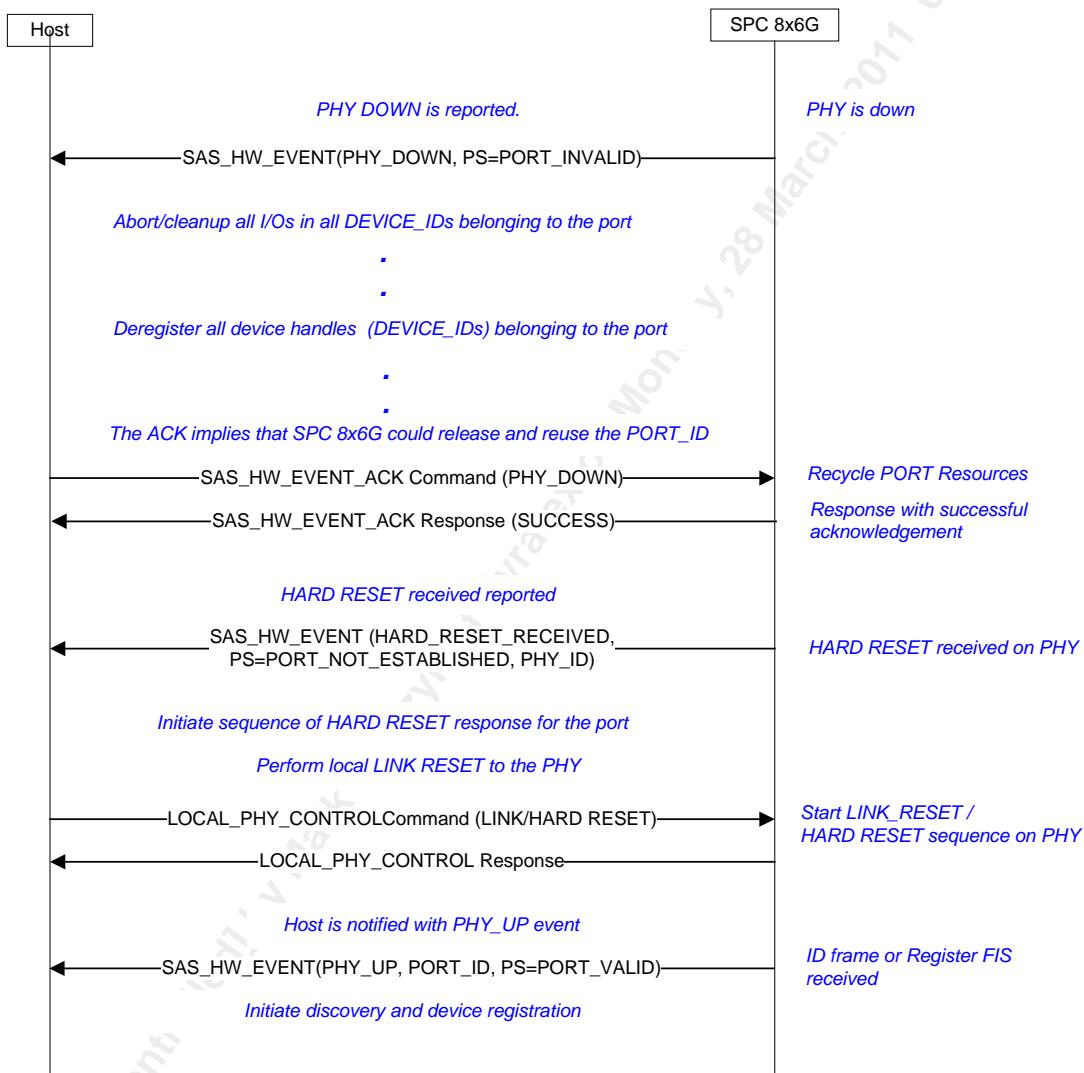


In a narrow port, when port recovery time is enabled, after the receipt of a **SAS_HW_EVENT Notification** (Section 8.2) with a `PHY_DOWN` event and a `HARD_RESET_RECEIVED` event, when the `PORT_ID` is still valid, the host can issue the **PORT_CONTROL Command** (Section 7.26) with a `HARD_RESET` operation to that `PORT_ID`. Note: There is an option in the **PORT_CONTROL Command** of using bit 0 of the `PARAM0` field to specify whether automatic device deregistration will be performed or not.

11.5.5.2 Narrow Port Hard Reset Received - Port Recovery Time Disabled

Figure 65 shows the handling of a narrow port Hard Reset with port recovery time disabled.

Figure 65 Narrow Hard Reset Received - Port Recovery Time Disabled

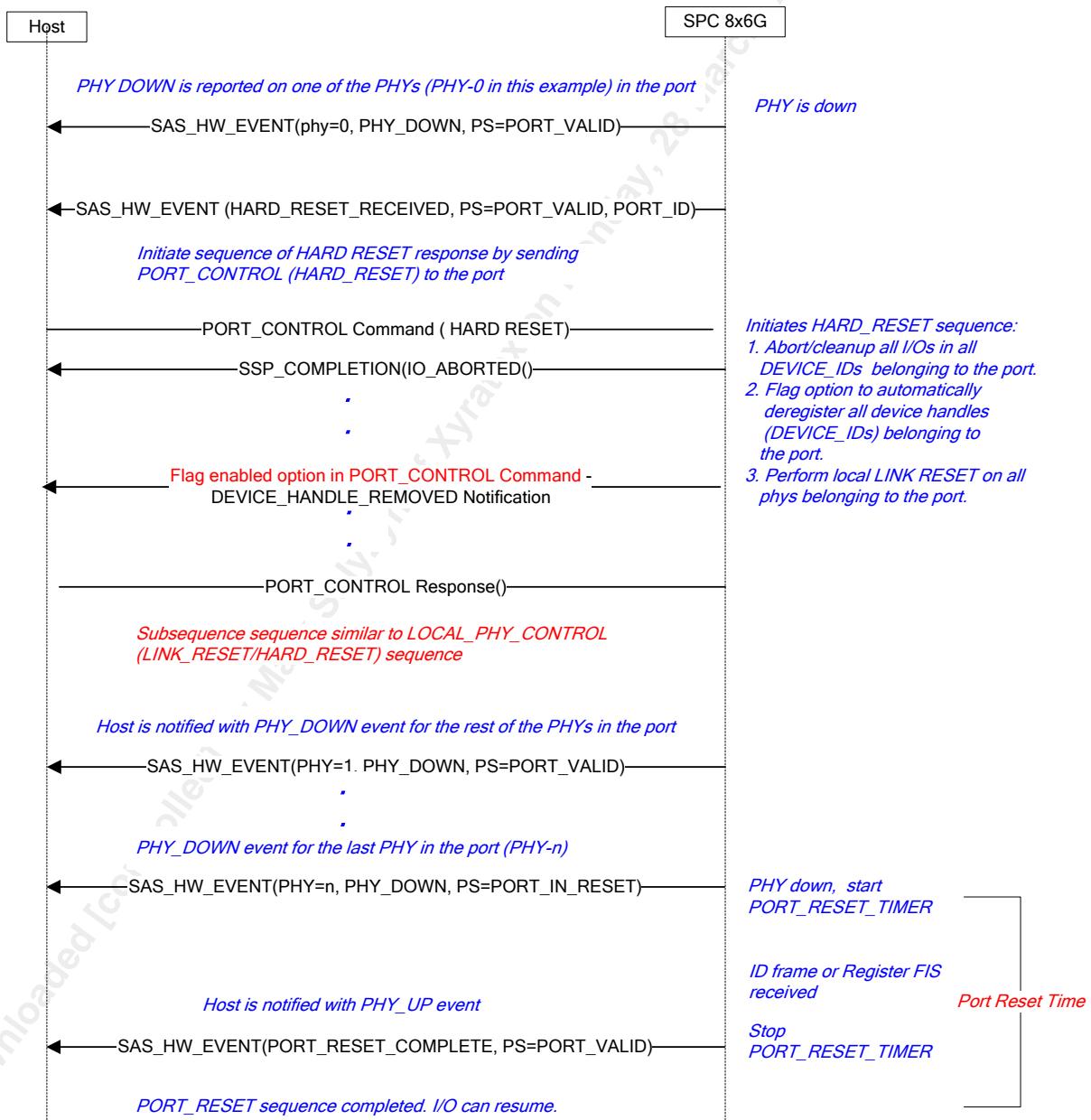


In a narrow port, when the port recovery time is disabled, after the receipt of a **SAS_HW_EVENT Notification** (Section 8.2) with the **PHY_DOWN** event and the **HARD_RESET_RECEIVED** event, and the **PORT_ID** is invalid, the host should issue the **LOCAL_PHY_CONTROL Command** (Section 7.18) with a **HARD_RESET** operation instead of the **PORT_CONTROL Command** (Section 7.26) with a **HARD_RESET** operation since the **PORT_ID** is not valid.

11.5.5.3 Wide Port Hard Reset Received – Port Recovery Time Enabled/Disabled

The figure below shows the handling of a wide port Hard Reset with the port recovery time enabled or disabled. In a wide port, there is no Hard Reset received handling distinction with the port recovery time enabled or disabled since the port is still valid when one of the PHYs is down.

Figure 66 Wide Port – Hard Reset Received – Port Recovery Time Enabled/Disabled



In a wide port, after the receipt of a [SAS_HW_EVENT Notification](#) (Section 8.2) with a PHY_DOWN event and a HARD_RESET RECEIVED event, the PORT_ID is still valid. The host can issue the [PORT_CONTROL Command](#) (Section 7.26) with a HARD_RESET operation to that PORT_ID. Note: There is an option in the [PORT_CONTROL Command](#) of using bit 0 of the PARAM0 field to specify whether automatic device deregistration will be performed or not.

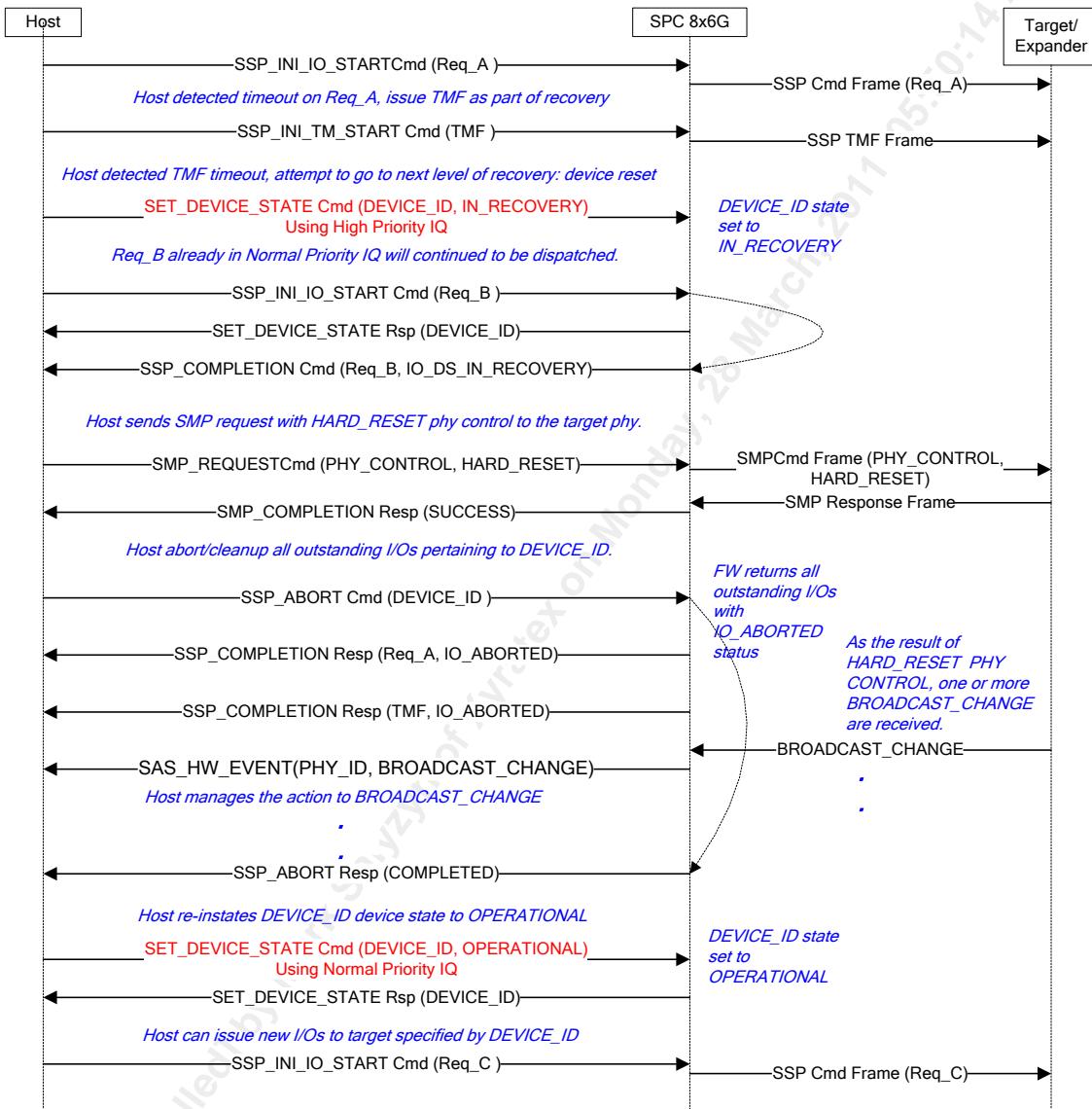
11.5.6 Device States Related Error Handler

This section describes the error recovery procedures involving the use of the device states specified by the DEVICE_ID. See [Table 8](#) in Section 3.2, “Device Handle and DEVICE_ID” and Section 7.29, “[SET_DEVICE_STATE Command](#)” for details about the device states.

11.5.6.1 Device Reset and SET_DEVICE_STATE Command

This section describes the direct use of the [SET_DEVICE_STATE Command](#) described in Section 7.29, as part of error recovery procedure that uses device reset.

Figure 67 Recovery with Device Reset and SET_DEVICE_STATE Command



As shown in Figure 67, the host sends the **SET_DEVICE_STATE Command** to change the device state to the DS_IN_RECOVERY state before initiating a device reset by sending the SMP PHY CONTROL to the device. The host sends the **SET_DEVICE_STATE Command** to set the device state to the DS_IN_RECOVERY state using the high priority IQ. At the completion of device reset error recovery sequence, the host sends the **SET_DEVICE_STATE Command** to set the device state to DS_OPERATIONAL using the normal priority IQ.

11.5.6.2 Task Management Special Control with Device State

This section describes the use of the DS and ADS options in the [SSP_INI_TM_START Command](#) to change the device state while sending a task management function in the initiator mode. (See Section 7.5, “[SSP_INI_TM_START Command](#)”.) The two cases:

- Task management function issued to a normal priority IQ and
- Task management function issued to a high priority IQ

are described in [Figure 68](#) and [Figure 69](#).

Figure 68 Task Management Recovery with DS=1 and Normal Priority IQ

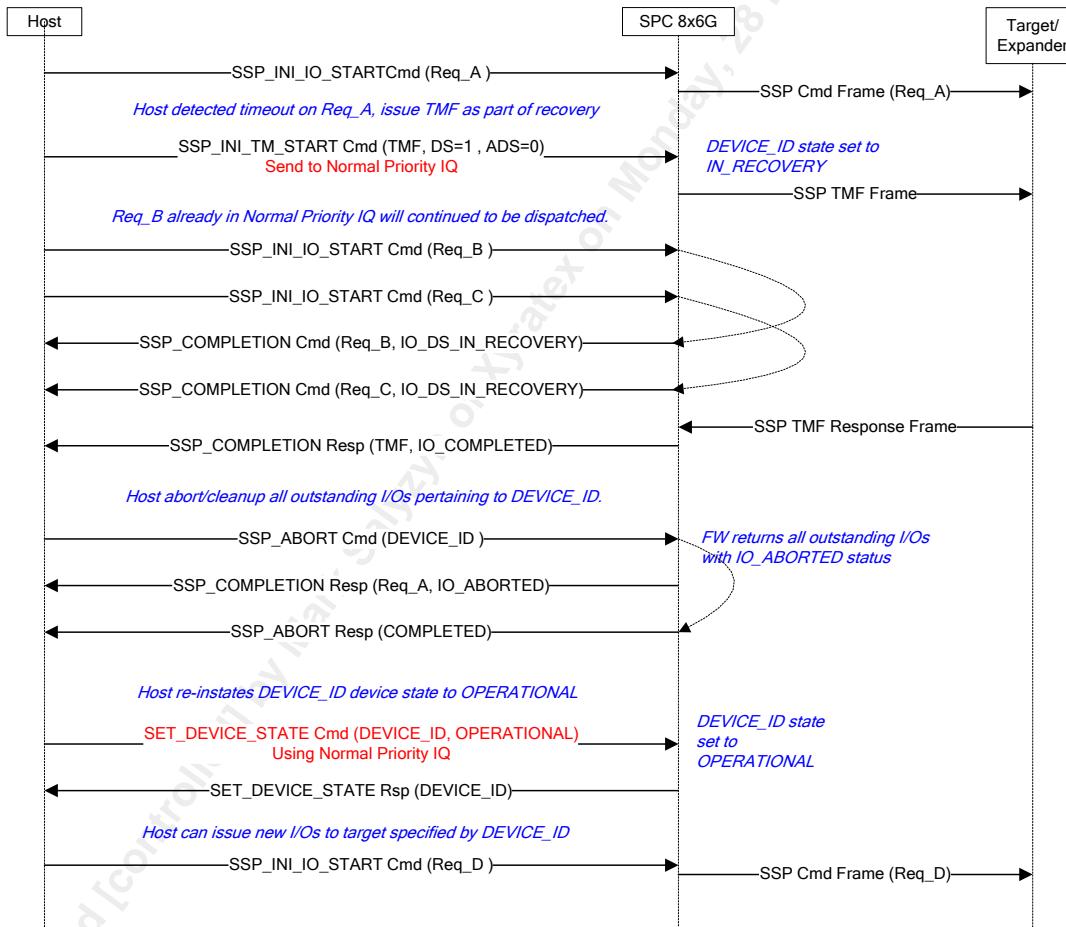
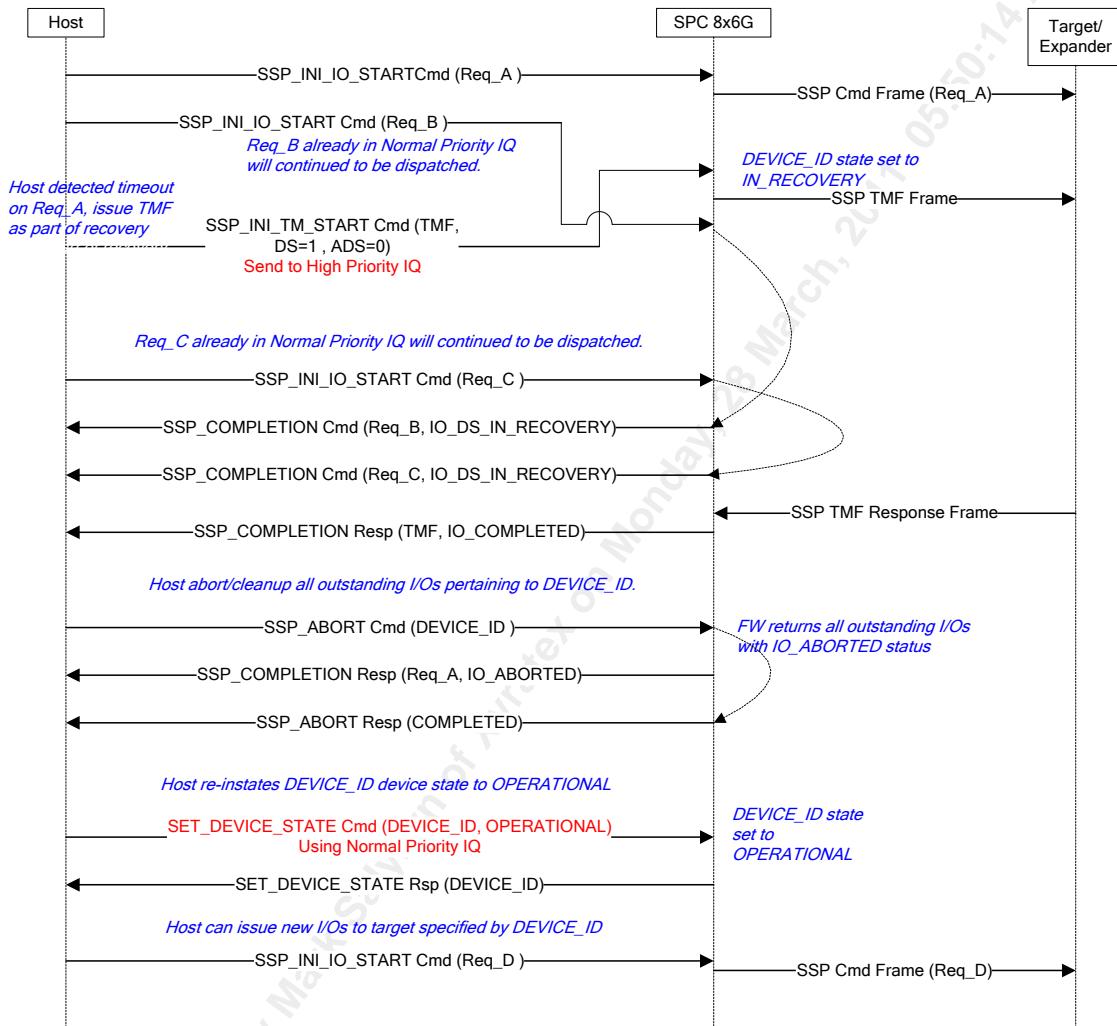
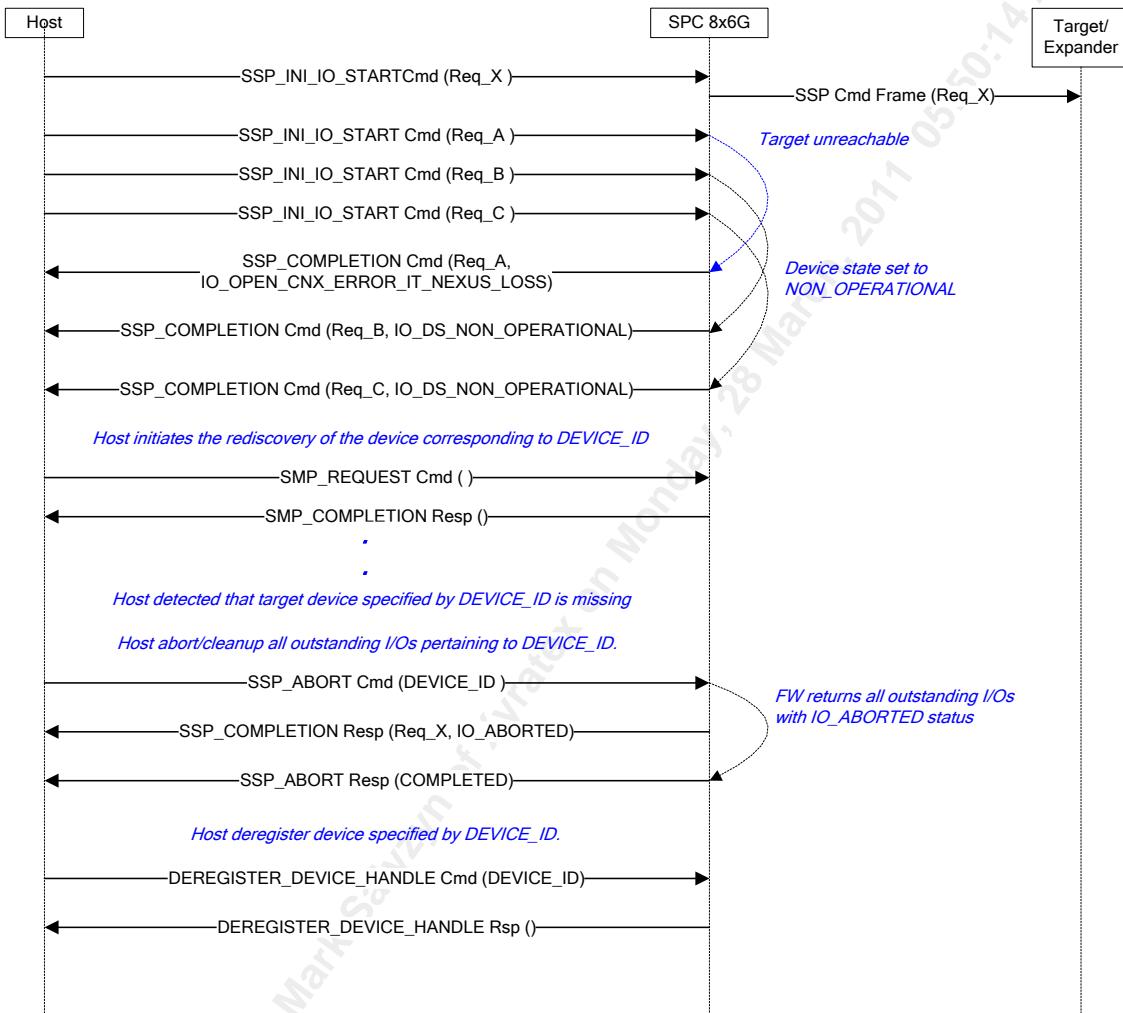


Figure 69 Task Management Recovery with DS=1 and High Priority IQ



11.5.6.3 Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices

For SSP operation, the IT_NEXUS_LOSS Timeout error is reported as an IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS (0x00000013) STATUS in the SSP_COMPLETION Response or the SSP_EVENT Notification. See Sections 8.3 and 8.11 respectively.

Figure 70 Head-Of-Line Blocking During IT_NEXUS_LOSS Timeout for SSP Devices


As shown in the above diagram, the I/O Req_B is completed in the **SSP_COMPLETION Response** with IO_DS_NON_OPERATIONAL set when the SPC 8x6G has changed the hardware device state to non-operational before the IT_NEXUS_LOSS timer for the Req_B command is started. If for a different timing reason the IT_NEXUS_LOSS timer for the Req_B command is already started before the SPC 8x6G has the chance to change the hardware device state to non-operational, the returned error status for Req_B will be IO_OPEN_CNX_ERROR_IT_NEXUS LOSS.

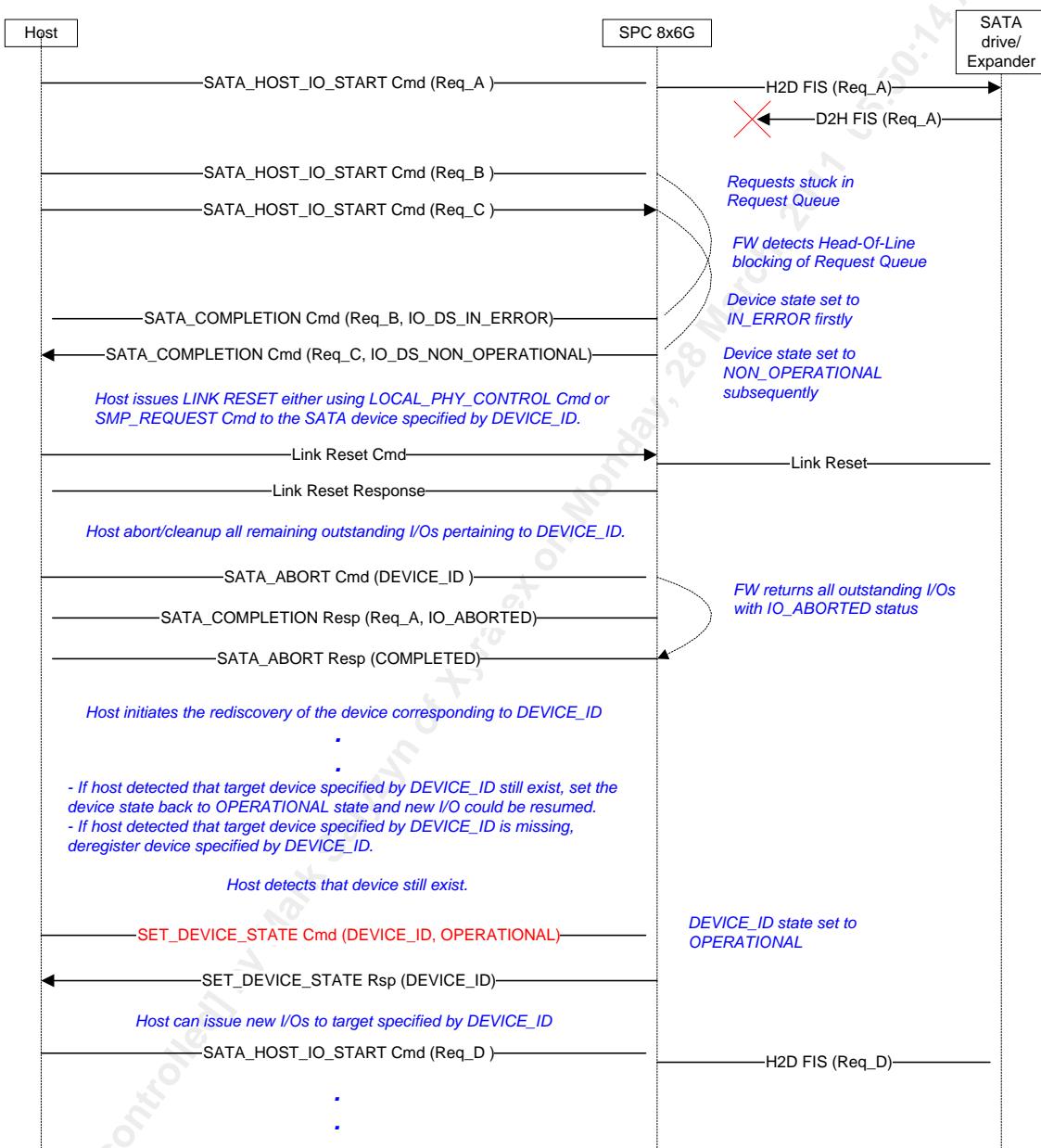
Also as shown in the diagram above, the path of using **SMP_REQUEST Command** to query or re-discover the devices behind the expanders is optional. The subsequent path to the **SSP_ABORT Command** with the DEVICE_ID parameter is still required even if the host decides not to de-register the device using the **DREGISTER_DEVICE_HANDLE Command**.

11.5.6.4 Head-Of-Line Blocking for SATA Protocol

The default timeout setting for the SPC 8x6G detection of SATA HOL blocking is 500 milliseconds. This value could be changed by setting the SATA_HOL_TMO value using the [SAS_RE_INITIALIZATION Command](#).

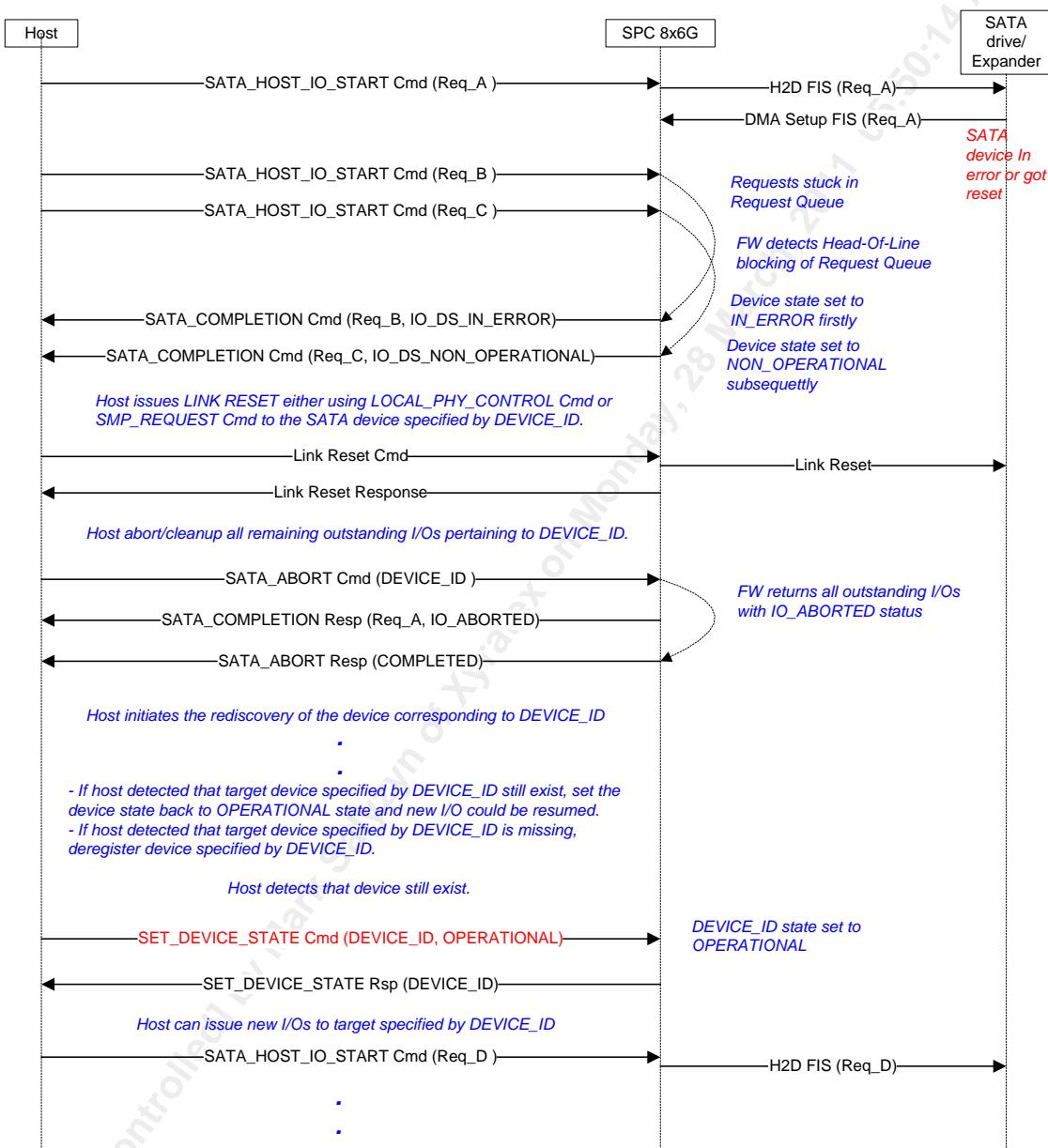
The following diagram shows the error handler in relation to head-of-line blocking of the SPC 8x6G internal request queue when performing a SATA NCQ operation (command) on the SATA drive.

Figure 71 Head-Of-Line Blocking for SATA Protocol (NCQ Command)



Similarly, the following diagram shows the error handler in relation to head-of-line blocking of the SPC 8x6G internal request queue when performing a SATA NCQ operation (data) on the SATA drive.

Figure 72 Head-Of-Line Blocking for SATA Protocol (NCQ Data)



11.6 Detailed Descriptions of the SAS Error Conditions

This section describes the SPC 8x6G SAS error conditions based on the layered model of SAS protocol:

- SAS physical layer
- SAS link layer
- SAS port layer
- SSP transport layer: initiator port
- SSP transport: target port
- SMP transport layer

11.6.1 SAS Physical Layer

11.6.1.1 SAS PHY Not Ready

The SPC 8x6G controller notifies the host with a [SAS_HW_EVENT Notification](#) (Section 8.2) with the EVENT field set to IOP_EVENT_PHY_ERR_PHY_RESET_FAILED when the PHY fails to complete speed negotiation. The host can directly read and reset the PHY reset failed counter using the [PHY Reset Failed Count Register](#) described in Section 10.6.10.

11.6.1.2 Invalid DWord

The SPC 8x6G controller detects invalid DWords both inside and outside frames. The host can directly read and reset the counter using the [Counter Configuration Register](#) described in Section 10.6.5.

For invalid DWords that are detected:

- During IDENTIFY frame reception — If an invalid DWord is detected after receiving a SOAF of the IDENTIFY frame and before receiving an EOAF, the SPC 8x6G controller increments the [Counter Configuration Register](#) and discards the frame. The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INVALID_DWORD. See Section 8.2. Eventually the IDENTIFY frame receive timeout is detected by the SPC 8x6G controller. See link layer error description for handling the IDENTIFY frame timeout in Section 11.6.2.2, “[IDENTIFY Frame Receive Timeout](#)”.
- During an OPEN address frame reception — If an invalid DWord is detected after receiving a SOAF of an OPEN address frame and before receiving an EOAF, the SPC 8x6G controller increments the [Counter Configuration Register](#) and discards the OPEN ADDRESS frame. The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INVALID_DWORD. See Section 8.2. Eventually this will result in an Open Timeout to the remote SAS port.

- During an SSP frame reception — If an invalid DWord is detected after receiving an SOF of an SSP frame and before receiving an EOF, the SPC 8x6G controller increments the [Counter Configuration Register](#), discards the frame, and sends a NAK (CRC ERROR) to the remote SAS device. The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INVALID_DWORD followed by another [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INBOUND_CRC. See Section 8.2.
- During an SMP frame reception — If an invalid DWord is detected after receiving an SOF of an SMP frame and before receiving an EOF, the SPC 8x6G controller increments the [Counter Configuration Register](#), detects an invalid SMP frame, and discards the frame. The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INVALID_DWORD followed by an [SMP_COMPLETION Response](#) with the STATUS field set to IO_ERROR_HW_TIMEOUT, if the SPC 8x6G detects a MAX_CONNECTION_TIMER expiration, or IO_XFER_ERR_BREAK, if a BREAK is received from the remote SMP target while the MAX_CONNECTION_TIMER is still running. See Sections 8.2 and 8.4. See the description of handling an invalid SMP frame in Section 11.6.2.3, “[Invalid SMP Frame](#)”.

11.6.1.3 SAS Running Disparity Error

The SPC 8x6G controller detects running disparity errors. The host can directly read and reset the counter using the [Disparity Error Count Register](#) described in Section 10.6.7.

The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_DISPARITY_ERROR. See Section 8.2. The occurrence of disparity errors may also cause the SPC 8x6G to detect invalid DWords. In this case, the host is notified with an additional [SAS_HW_EVENT Notification](#) as described in Section 11.6.1.2, “[Invalid DWord](#)”.

11.6.1.4 Loss of DWord Synchronization

The SPC 8x6G controller detects a loss of DWord synchronization when the PHY is in READY state. The host can directly read and reset the error counter using the [Loss of DWord Synchronization Count Register](#) described in Section 10.6.9.

The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_LOSS_OF_DWORD_SYNCH. The loss of DWord synchronization causes the SPC 8x6G to initiate the PHY link reset sequence automatically. The host is also notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_DOWN. See Section 11.5.1, “[PHY Down Handler \(External Trigger\)](#)” for details.

11.6.1.5 PHY Reset Problems

The SPC 8x6G controller handles PHY reset problems as described in Section 11.6.1.1, “[SAS PHY Not Ready](#)”.

11.6.1.6 Elasticity Buffer Overflow

This condition will not occur for the SPC 8x6G.

11.6.1.7 ERROR Primitive

The SPC 8x6G controller detects ERROR primitives that are received on the links. An ERROR primitive inside a frame is considered to be a CRC error. The host can use the [SAS_DIAG_EXECUTE Command](#) described in Section 7.23 to program and retrieve the ERROR primitive receive count.

11.6.2 SAS Link Layer

11.6.2.1 CRC Errors

The SPC 8x6G controller detects frame CRC errors and notifies the host. A CRC error may lead to additional error notifications depending on the frame type that has the error.

For CRC errors:

- During an IDENTIFY frame reception — If the CRC error is detected after receiving an SOAF of an IDENTIFY frame and before receiving an EOAF, the SPC 8x6G controller discards the frame. The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INBOUND_CRC. Eventually the IDENTIFY frame receive timeout is detected by the SPC 8x6G controller. See the link layer error description in Section 11.6.2.2, “[IDENTIFY Frame Receive Timeout](#)” for how to handle an IDENTIFY frame timeout.
- During an OPEN address frame reception — If the CRC error is detected after receiving an SOAF of an OPEN address frame and before receiving an EOAF, the SPC 8x6G controller discards the OPEN address frame. The host is notified with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INBOUND_CRC.

Eventually, the remote SAS device detects an OPEN timeout and retries the OPEN connection request.

- During an SSP frame reception — If the CRC error is detected after receiving an SOF of an SSP frame and before receiving of an EOF, the SPC 8x6G controller treats the frame as invalid, discards the frame, and sends a NAK (CRC ERROR) to the remote SAS device.

The SPC 8x6G controller notifies the host with a [SAS_HW_EVENT Notification](#) of outbound IOMB with a SAS EVENT set to IOP_EVENT_PHY_ERR_INBOUND_CRC. See Section 8.2 for the full description of a [SAS_HW_EVENT Notification](#) of outbound IOMB.

If the discarded frame is a data frame and a subsequent data frame is received for the same tag, the SPC 8x6G controller notifies the host with an [SSP_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_OFFSET_MISMATCH. See Section 8.11.

If the discarded frame is a data frame and a response frame is received for the same tag, the SPC 8x6G controller completes the command with an outbound [SSP_COMPLETION Response](#) with the STATUS field set to IO_XFER_ERROR_DMA. See Section 8.3.

- During an SMP frame reception — If the CRC error is detected after receiving an SOF of an SMP frame and before receiving an EOF, the SPC 8x6G controller detects the invalid SMP frame and discards it.

The SPC 8x6G controller notifies the host with an [SAS_HW_EVENT Notification](#) of outbound IOMB with a SAS EVENT set to IOP_EVENT_PHY_ERR_INBOUND_CRC. See Section 8.2. Subsequently, the SPC 8x6G controller notifies the host with an outbound [SMP_COMPLETION Response](#) with the STATUS set to IO_XFER_ERROR_HW_TIMEOUT. See Section 8.4.

11.6.2.2 IDENTIFY Frame Receive Timeout

When the SPC 8x6G controller fails to receive (timeout) an IDENTIFY frame following the PHY reset sequence, the controller notifies the host with an [SAS_HW_EVENT Notification](#) with the SAS EVENT set to IOP_EVENT_ID_FRAME_TIMEOUT. See Section 8.2 for details.

The SPC 8x6G controller automatically restarts the PHY link reset sequence following an IDENTIFY frame timeout. The retries are indefinite.

11.6.2.3 Invalid SMP Frame

If the SPC 8x6G controller receives an invalid SMP response frame (the controller in SMP initiator mode), it discards the frame, transmits a BREAK to end the connection, and notifies the host by an IOMB outbound [SMP_COMPLETION Response](#) (Section 8.4) with the STATUS set to IO_XFER_ERROR_RX_FRAME. The host can retry the SMP request later.

11.6.2.4 Invalid SSP Frame

An SSP frame that has any of the following properties is considered an invalid frame and is dropped by the SPC 8x6G controller:

- Frame length is greater than 263 DWords
- Frame length is less than 7 DWords
- Frame is received after the CREDIT is exhausted
- Frame is received after receiving is DONE
- Frame without an EOF primitive
- Frame with ERROR primitive

The subsequent action depends on the remote SAS device behavior. If no other frames are received by the SPC 8x6G from the remote device in question, the host will detect a ULP timeout for one or more of its SSP operations.

11.6.2.5 Open Connection Errors

Open Connection Timeout: If the SPC 8x6G controller does not get the response for an OPEN request when the Open Connection timer expires, it sends a BREAK to abort the open request and retries the open until IT_Nexus_Loss timer expires. See the port layer error description in Section 11.6.3.1, “[IT Nexus Loss Timeout](#)” for details about handling IT_Nexus_Loss.

OPEN_REJECT Received: The receipt of an OPEN_REJECT specifies that the connection request is rejected and the reason for the rejection. An OPEN_REJECT received may result in abandonment of the connection request or a retry of the open request. See the primitives for each in [Table 370](#) and [Table 371](#).

Table 370 Abandon Class OPEN_REJECT Primitives

OPEN_REJECT Version	Description
PROTOCOL NOT_SUPPORTED ZONE VIOLATION BAD DESTINATION RATE NOT SUPPORTED PROTOCOL NOT SUPPORTED WRONG DESTINATION STP RESOURCES BUSY	<p>When the SPC 8x6G controller receives any of the abandon-class OPEN_REJECT primitives, it does not retry the open request.</p> <p>For SSP:</p> <p>If the connection request is rejected during the data phase, the SPC 8x6G controller notifies the host with an SSP_EVENT Notification with the EVENT field set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_ZONE_VIOLATION • IO_OPEN_CNX_ERROR_BAD_DESTINATION • IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_WRONG_DESTINATION <p>See Section 8.11, “SSP_EVENT Notification” for details.</p> <p>If the connection request is rejected during the command phase, the SPC</p>

OPEN_REJECT Version	Description
	<p>8x6G controller notifies the host with an SSP_COMPLETION Response with the STATUS set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_ONE_VIOLATION • IO_OPEN_CNX_ERROR_BAD_DESTINATION • IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_WRONG_DESTINATION <p>See Section 8.3, "SSP_COMPLETION Response" for details.</p> <p>For SMP:</p> <p>The SPC 8x6G controller notifies the host with an SMP_COMPLETION Response with its STATUS set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_ONE_VIOLATION • IO_OPEN_CNX_ERROR_BAD_DESTINATION • IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_WRONG_DESTINATION <p>See Section 8.4, "SMP_COMPLETION Response" for details.</p> <p>For STP:</p> <p>If the connection request is rejected during the data phase, the SPC 8x6G notifies the host with a SATA_EVENT Notification with the EVENT field set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_ZONE_VIOLATION • IO_OPEN_CNX_ERROR_BAD_DESTINATION • IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_WRONG_DESTINATION • IO_OPEN_CNX_ERROR_STP_RESOURCES_BUSY <p>See Section 8.10, "SATA_EVENT Notification" for details.</p> <p>If the connection request is rejected during the command phase, the SPC 8x6G controller notifies the host with a SATA_COMPLETION Response with the STATUS set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_PROTOCOL_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_ONE_VIOLATION • IO_OPEN_CNX_ERROR_BAD_DESTINATION • IO_OPEN_CNX_ERROR_CONNECTION_RATE_NOT_SUPPORTED • IO_OPEN_CNX_ERROR_WRONG_DESTINATION <p>See Section 8.9, "SATA_COMPLETION Response" for details.</p>

OPEN_REJECT Version	Description
	<p>When the SPC 8x6G receives an OPEN_REJECT(STP_RESOURCES_BUSY), the host is notified with a SATA_COMPLETION Response with STATUS set to IO_OPEN_CNX_ERROR_STP_RESOURCES_BUSY. See Section 8.9, “SATA_COMPLETION Response” for details.</p> <p>The host may choose to retry the command after clearing the affliction by sending an SMP_PHY_CONTROL function.</p>

Table 371 Retry Class OPEN_REJECT Primitives

OPEN_REJECT Version	Description
NO DESTINATION PATHWAY BLOCKED RETRY	<p>When the SPC 8x6G controller receives any of the retry-class OPEN_REJECT primitives, it retries the connection request.</p> <p>For NO DESTINATION and PATHWAY BLOCKED retry primitives, the connection request is retried until the IT_Nexus_Loss timer is expired. After the timer is expired the host is notified with the error code, IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. The IT_NEXUS_LOSS time is set per device during device registration. See Section 11.6.3.1, “IT Nexus Loss Timeout” for details.</p> <p>For the RETRY primitive, the connection request is retried until the internal retry counter expires. By default, the SPC 8x6G retries 2048 times. Please see Section 7.32, “SAS RE_INITIALIZATION Command”. After the timer is expired the host is notified with the error code, IO_XFER_OPEN_RETRY_TIMEOUT.</p> <p>For SSP:</p> <p>If the connection request is rejected during the data phase, the SPC 8x6G notifies the host with an SSP_EVENT Notification with EVENT set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS • IO_XFER_OPEN_RETRY_TIMEOUT <p>See Section 8.11, “SSP_EVENT Notification” for details.</p> <p>If the connection request is rejected during the command phase, the SPC 8x6G notifies the host with an SSP_COMPLETION Response with STATUS set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS • IO_XFER_OPEN_RETRY_TIMEOUT <p>See Section 8.3, “SSP_COMPLETION Response” for details.</p> <p>For SMP:</p> <p>The SPC 8x6G controller notifies the host with an SMP_COMPLETION Response with STATUS set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS • IO_XFER_OPEN_RETRY_TIMEOUT <p>See Section 8.4, “SMP_COMPLETION Response” for details.</p> <p>For STP:</p> <p>If the connection request is rejected during the data FIS phase, the SPC 8x6G notifies the host with a SATA_EVENT Notification with EVENT set</p>

OPEN_REJECT Version	Description
	<p>to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS • IO_XFER_OPEN_RETRY_TIMEOUT <p>See Section 8.10, "SATA_EVENT Notification" for more details.</p> <p>If the connection request is rejected during the H2D FIS phase, the SPC 8x6G controller notifies the host with a SATA_COMPLETION Response with STATUS set to one of the following depending on the OPEN_REJECT version:</p> <ul style="list-style-type: none"> • IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS • IO_XFER_OPEN_RETRY_TIMEOUT <p>See Section 8.9, "SATA_COMPLETION Response" for more details.</p>

11.6.2.6 BREAK Received During Open Request

A connection request may fail if a BREAK is received in response to an OPEN request. When the SPC 8x6G controller detects a BREAK, the open request is aborted and the host is notified.

In initiator mode:

For SSP, the SPC 8x6G controller notifies the host with an SSP_EVENT Notification the STATUS set to IO_OPEN_CNX_ERROR_BREAK or IO_XFER_ERROR_BREAK. See Section 8.11, "[SSP_EVENT Notification](#)" and Section 8.3 "[SSP_COMPLETION Response](#)" for details.

- During the SSP command phase, the SPC 8x6G controller notifies the host with SSP_COMPLETION Response outbound IOMB with its STATUS field set to IO_OPEN_CNX_ERROR_BREAK.
- During the SSP data phase, the SPC 8x6G controller notifies the host with an SSP_EVENT Notification with the EVENT field set to IO_XFER_ERROR_BREAK or IO_OPEN_CNX_ERROR_BREAK.

For SMP, the SPC 8x6G controller notifies the host with an SMP_COMPLETION Response with STATUS set to IO_OPEN_CNX_ERROR_BREAK. See Section 8.4 for details.

For STP, the SPC 8x6G controller notifies the host with a SATA_EVENT Notification with STATUS set IO_OPEN_CNX_ERROR_BREAK or IO_XFER_ERROR_BREAK

- For the STP H2D phase, the SPC 8x6G controller notifies the host with a SATA_COMPLETION response outbound IOMB with STATUS set to IO_OPEN_CNX_ERROR_BREAK.
- For the STP Data phase, the SPC 8x6G controller notifies the host with an SATA_EVENT Notification with the EVENT field set to IO_XFER_ERROR_BREAK or IO_OPEN_CNX_ERROR_BREAK

In target mode:

For SSP, the SPC 8x6G controller notifies the host with an [SSP_COMPLETION Response](#) the STATUS set to IO_OPEN_CNX_ERROR_BREAK or IO_XFER_ERROR_BREAK. See [Section 8.3, “SSP_COMPLETION Response”](#) for details.

11.6.2.7 BROADCAST Received

When the SPC 8x6G controller receives a BROADCAST from a PHY, it notifies the host.

The SPC 8x6G controller notifies the host with a [SAS_HW_EVENT Notification](#) outbound IOMB with its SAS EVENT set to IOP_EVENT_BROADCAST_CHANGE (0x00000009) or IOP_EVENT_BROADCAST_SES (0x0000000b). See [Section 8.2](#) for the full description of the [SAS_HW_EVENT Notification](#) outbound IOMB.

For a BROADCAST (CHANGE), the host may need to do re-discovery. For BROADCAST (SES), the host may need to poll each enclosure service device to find out the reason.

11.6.2.8 HARD_RESET Received

The SPC 8x6G controller notifies the host with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_HARD_RESET_RECEIVED when it receives a HARD_RESET primitive on one of its PHYs. Since receipt of a HARD_RESET is always preceded by a link reset sequence, the SPC 8x6G notifies the host with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_DOWN prior to the notification being sent with the EVENT field set to IOP_EVENT_HARD_RESET_RECEIVED. See [Section 8.2](#) for the full description of the [SAS_HW_EVENT Notification](#) outbound IOMB.

The SPC 8x6G freezes the port containing the PHY receiving the HARD_RESET to pause all the ongoing traffic from the port. The host should initiate a link reset to all the PHYs that make up a port by sending a [LOCAL_PHY_CONTROL Command](#) ([Section 7.18](#)) with PHY OP set to LINK RESET (0x02) or by sending a [PORT_CONTROL Command](#) ([Section 7.26](#)) with PORT OP set to HARD RESET. The host is responsible to abort all pending I/Os to the devices connected via that port and optionally deregister all the devices. See [Section 11.5.5, “Hard Reset Received Handler”](#) for further details.

In initiator mode, the host could reinitiate the I/Os after receiving a [LOCAL_PHY_CONTROL Response](#) outbound IOMB with its STATUS set to SUCCESS.

In target mode, the host will receive an event when the initiator tries to connect.

11.6.2.9 SSP NAK Received

The SPC 8x6G implements Transport Layer Retry (TLR) as specified by the SAS-1.0 and SAS-2 specifications. See [Section 3.20, “Transport Layer Retry \(TLR\) Handling”](#).

As an initiator, when the SPC 8x6G receives a NAK to a COMMAND that it has sent:

- If TLR is enabled, the SPC 8x6G will retry the COMMAND until it succeeds or the TLR counter is exhausted. If the SPC 8x6G continues receiving NAKs after the counter expires, it will terminate the I/O and will notify the host with an [SSP_COMPLETION Response](#) (Section 8.3) outbound IOMB with its STATUS set to IO_XFER_ERROR_NAK RECEIVED.
- If TLR is disabled, the SPC 8x6G will immediately terminate the I/O and will notify the host with an [SSP_COMPLETION Response](#) outbound IOMB with its STATUS set to IO_XFER_ERROR_NAK RECEIVED.

As an initiator, when the SPC 8x6G receives a NAK to a DATA_OUT frame servicing an XFER_RDY:

- If TLR is enabled, the SPC 8x6G will retry the DATA_OUT phase provided that the target-enabled TLR support and the TLR counter are not exhausted. If the SPC 8x6G continues receiving NAKs after the counter expires, it will terminate the I/O and will notify the host with an [SSP_EVENT Notification](#) outbound IOMB with the EVENT field set to IO_XFER_ERROR_NAK RECEIVED. The subsequent events that may follow depend on the target.
- If TLR is disabled, the SPC 8x6G will notify the host with an [SSP_EVENT Notification](#) outbound IOMB with the EVENT field set to IO_XFER_ERROR_NAK RECEIVED. The subsequent events that may follow depend on the target.

As a target, when the SPC 8x6G receives a NAK to a DATA/XFER_RDY/RESPONSE frame:

- If TLR is enabled, the SPC 8x6G will retry the phase until the TLR counter is exhausted. If the SPC 8x6G continues receiving NAKs after the counter expires, it will terminate the I/O and will notify the host with an [SSP_COMPLETION Response](#) outbound IOMB with the STATUS field set to IO_XFER_ERROR_NAK RECEIVED. The subsequent events that may follow depend on the target host application.
- If TLR is disabled, the SPC 8x6G will terminate the I/O and will notify the host with an [SSP_COMPLETION Response](#) outbound IOMB with the STATUS field set to IO_XFER_ERROR_NAK RECEIVED. The subsequent events that may follow depend on the target host application.

See Sections 8.3, “[SSP_COMPLETION Response](#)” and 8.11, “[SSP_EVENT Notification](#)” for details.

11.6.2.10 ACK/NAK Timeouts

The SPC 8x6G implements Transport Layer Retry (TLR) as specified by the SAS-1.0 and SAS-2 specifications. See Section 3.20 for more details.

The SPC 8x6G ACK/NAK timeout timer is set to 800 us. This is implemented so that the local ACK/NAK timer will expire before the remote DONE (ACK/NAK Timeout) expires in the case of a missing ACK. This ensures that the SSP connection gets closed with a DONE (ACK/NAK timeout) instead of a BREAK.

As an initiator, when the SPC 8x6G detects an ACK/NAK timeout to a COMMAND or TASK frame that it has sent, regardless of the TLR setting, the SPC 8x6G will notify the host with an [SSP_EVENT Notification](#) outbound IOMB (Section 8.3) with its EVENT set to IO_XFER_ERROR_CMD_ISSUE_ACK_NAK_TIMEOUT. The host may need to send a QUERY_TASK (task management) to determine whether the remote SAS device has received the COMMAND or the TASK.

As an initiator, when the SPC 8x6G detects an ACK/NAK timeout to a DATA frame that it has sent:

- If TLR is enabled, the SPC 8x6G will retry the DATA_OUT phase provided that the target-enabled TLR support and the TLR counter are not exhausted. If the SPC 8x6G continues detecting ACK/NAK timeout conditions after the counter expires, it will terminate the I/O and will notify the host with an [SSP_EVENT Notification](#) outbound IOMB with the EVENT field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT. The subsequent events that may follow depend on the target.
- If TLR is disabled, the SPC 8x6G will notify the host with an [SSP_EVENT Notification](#) outbound IOMB with the EVENT field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT. The subsequent events that may follow depend on the target.

As a target, when the SPC 8x6G detects an ACK/NAK timeout to a DATA frame that it has sent:

- If TLR is enabled, the SPC 8x6G will retry the DATA_OUT phase until the TLR counter is exhausted. If the SPC 8x6G continues detecting ACK/NAK timeout conditions after the counter expires, it will terminate the I/O and will notify the host with an [SSP_COMPLETION Response](#) outbound IOMB with the STATUS field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT. The subsequent events that may follow depend on the host application.
- If TLR is disabled, the SPC 8x6G will notify the host with an [SSP_COMPLETION Response](#) outbound IOMB with the STATUS field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT. The subsequent events that may follow depend on the host application.

As a target, when the SPC 8x6G detects an ACK/NAK timeout to a RESPONSE frame or XFER_RDY frame that it has sent:

- If TLR is enabled, the SPC 8x6G will retry the frame until the TLR counter is exhausted. If the SPC 8x6G continues detecting ACK/NAK timeout conditions after the counter expires, it will terminate the I/O and will notify the host with [SSP_COMPLETION Response](#) outbound IOMB with the STATUS field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT. The subsequent events that may follow depend on the host application.
- If TLR is disabled, the SPC 8x6G will notify the host with an [SSP_COMPLETION Response](#) outbound IOMB with the STATUS field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT. The subsequent events that may follow depend on the host application.

See Section 8.11, “[SSP_EVENT Notification](#)” and Section 8.3 “[SSP_COMPLETION Response](#)” for details.

11.6.2.11 BREAK Received During Connection

If the SPC 8x6G controller detects a BREAK during a connection, the SPC 8x6G controller aborts the operation and notifies the host:

- During the SSP command phase, the SPC 8x6G controller notifies the host with an [SSP_EVENT Notification](#) outbound IOMB with its STATUS field set to IO_XFER_ERROR_BREAK if the BREAK is received before the ACK/NAK during command frame transmission. The SPC 8x6G ignores the BREAK that is received after receiving the ACK for the command frame.
- During the SSP data phase, the SPC 8x6G controller notifies the host with an [SSP_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_BREAK or IO_OPEN_CNX_ERROR_BREAK if the BREAK field is received before the ACK/NAK during data frame transmission. The SPC 8x6G ignores the BREAK that is received after receiving the ACK for the data frame if ACK/NAK balance is achieved. If the connection is closed without ACK/NAK balance, the SPC 8x6G notifies the host with an [SSP_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_ACK_NAK_TIMEOUT.
- For SMP, the SPC 8x6G controller notifies the host with an [SMP_COMPLETION Response](#) outbound IOMB with its STATUS set to IO_XFER_ERROR_BREAK if the BREAK is received during SMP request frame transmission.
- For the STP H2D phase, the SPC 8x6G controller notifies the host with a [SATA_EVENT Notification](#) outbound IOMB with STATUS set to IO_XFER_ERROR_BREAK if the BREAK is received before R_ERR during H2D FIS transmission. The SPC 8x6G controller ignores the BREAK that is received after receiving an R_OK for the H2D FIS.
- For the STP Data phase, the SPC 8x6G controller notifies the host with an [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_BREAK or IO_OPEN_CNX_ERROR_BREAK if the BREAK is received before an R_OK/R_ERR during data FIS transmission. The SPC 8x6G ignores the BREAK that is received after receiving an R_OK for the data frame.

See Section 8.3, “[SSP_COMPLETION Response](#)” for details, Section 8.11, “[SSP_EVENT Notification](#)”, Section 8.4, “[SMP_COMPLETION Response](#)”, Section 8.9, “[SATA_COMPLETION Response](#)”, Section 8.10, “[SATA_EVENT Notification](#)” for details.

11.6.2.12 CREDIT Timeout During Connection

If the SPC 8x6G controller does not receive a CREDIT from the remote SAS device, it closes the connection by sending DONE(CREDIT_TIMEOUT) and reopens the connection. This sequence occurs indefinitely. At some point, the host may detect a command timeout if the remote SAS device fails to send a CREDIT.

11.6.2.13 PHY Not Ready During Connection

The SPC 8x6G controller detects link down connections during frame transmission. It aborts the operation and notifies the host with an event or completion depending on the data phase:

- For the SSP command phase, the SPC 8x6G controller notifies the host with an [SSP_COMPLETION Response](#) outbound IOMB with STATUS set to IO_XFER_ERROR_PHY_NOT_READY.
- For the SSP data/XFR_RDY/Response phase, the SPC 8x6G controller notifies the host with an [SSP_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_PHY_NOT_READY.
- For SMP, the SPC 8x6G controller notifies the host with an [SMP_COMPLETION Response](#) outbound IOMB with STATUS set to IO_XFER_ERROR_PHY_NOT_READY.
- For STP H2D FIS, the SPC 8x6G controller notifies the host with a [SATA_COMPLETION Response](#) outbound IOMB with STATUS set to IO_XFER_ERROR_PHY_NOT_READY.
- For STP data FIS, the SPC 8x6G controller notifies with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_PHY_NOT_READY.

See Section 8.3, “[SSP_COMPLETION Response](#)”, Section 8.11, “[SSP_EVENT Notification](#)”, Section 8.4, “[SMP_COMPLETION Response](#)”, Section 8.9, “[SATA_COMPLETION Response](#)”, and Section 8.10, “[SATA_EVENT Notification](#)” for details.

11.6.3 SAS Port Layer

11.6.3.1 I_T Nexus Loss Timeout

If the SPC 8x6G controller detects an I_T Nexus Loss timeout during an open connection, it notifies the host and it may or may not abort the operation.

The SPC 8x6G controller notifies the host with an [SSP_COMPLETION Response](#) or an [SSP_EVENT Notification](#) outbound IOMB with its STATUS set to IO_OPEN_CNX_ERROR_IT_NEXUS_LOSS. See Section 8.3, “[SSP_COMPLETION Response](#)” for further details. If the I_T Nexus Loss is detected when sending the command, the SPC 8x6G terminates the I/O with an [SSP_COMPLETION Response](#). If the I_T Nexus Loss is detected in other phases, the SPC 8x6G notifies the host with an [SSP_EVENT Notification](#) and the command may still be pending at the target. See Section 8.11 for the full description of the [SSP_EVENT Notification](#) outbound IOMB.

The host can retry the request later.

11.6.4 SSP Transport Layer (Initiator Port)

11.6.4.1 COMMAND/TASK Frame Received

If the SPC 8x6G SSP initiator port receives a COMMAND or a TASK frame, the controller discards the frame.

In order for it to receive a COMMAND or TASK frame, the host should configure the SPC 8x6G controller to support SSP target mode.

11.6.4.2 Vendor-specific SSP Frame Received

If the SPC 8x6G SSP initiator port receives a vendor-specific SSP frame type, the controller discards the frame.

In order for it to receive vendor specific SSP frame types, the host should configure the controller to support SSP target mode.

11.6.4.3 XFER_RDY/DATA/RESPONSE Frame Received with Unknown Tag

If the SPC 8x6G SSP initiator port receives an XFER_RDY or DATA or RESPONSE frame with an unknown tag, the controller ignores it (drop the frame).

11.6.4.4 Unexpected XFER_RDY Received

If the SPC 8x6G SSP initiator port receives an XFER_RDY frame with a command tag when it is not expecting one, the controller discards it and notifies the host with an [SSP_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_XFER_RDY_NOT_EXPECTED as described in Section [8.11](#).

The host must perform recovery operation on the tag before retrying the IO request.

11.6.4.5 Invalid XFER_RDY Received

If the SPC 8x6G SSP initiator port receives an invalid XFER_RDY frame with one of the following errors, the controller discards the frame, aborts the command, and notifies the host with one of the error messages shown in [Table 372](#). The controller notifies the host with an [SSP_EVENT Notification](#) with EVENT set as shown in [Table 372](#).

Table 372 SSP Initiator Invalid XFER_RDY Received

Error Message	EVENT Setting
Information Unit too long	The SPC 8x6G silently drops the packet
Information Unit too short	The SPC 8x6G silently drops the packet
Incorrect write data length	IO_XFER_ERROR_XFER_RDY_OVERRUN
Offset error	IO_XFER_ERROR_OFFSET_MISMATCH

See Section 8.11, “[SSP_EVENT Notification](#)” for details.

11.6.4.6 Unexpected DATA Frame Received

If the SPC 8x6G SSP initiator port receives a DATA frame with a command tag, when it is not expecting one, the controller discards the frame and notifies the host with an [SSP_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_UNEXPECTED_PHASE as described in Section [8.11](#).

The host must perform recovery operation on the tag before retrying the IO request.

11.6.4.7 Invalid DATA Frame Received

If the SPC 8x6G SSP initiator port receives an invalid DATA frame with one of the following errors, the controller discards the frame, aborts the operation, and notifies the host with the error message as shown in [Table 373](#).

Table 373 SSP Initiator Invalid DATA Frame Received

Error Message	EVENT Setting
Too much read data (command overrun)	The SPC 8x6G notifies the host with an SSP_EVENT Notification with the EVENT field set to IO_OVERFLOW. Subsequently, if a response frame is received, the SPC 8x6G reports an outbound SSP_COMPLETION Response with the STATUS field set to IO_XFER_ERROR_DMA
Information Unit too short	The SPC 8x6G notifies the host with an SSP_COMPLETION Response with the STATUS field set to IO_UNDERFLOW if the response is received before satisfying the DATA length specified for the command. The DATA frame is considered valid if it contains a complete frame header; otherwise, the packet will be silently dropped
Offset error	The SPC 8x6G notifies the host with an SSP_EVENT Notification with the EVENT field set to IO_XFER_ERROR_OFFSET_MISMATCH. Subsequently, if the response frame is received, the SPC 8x6G reports an SSP_COMPLETION Response with the STATUS field set to IO_XFER_ERROR_DMA.

See Sections [8.3](#), “[SSP_COMPLETION Response](#)” and [8.11](#) for details “[SSP_EVENT Notification](#)”.

11.6.4.8 Command Underrun

If the SPC 8x6G SSP initiator port receives a good RESPONSE frame with a GOOD SCSI status for a READ command before all the data for the READ command is received, a command underrun/underflow occurs.

The controller notifies the host with an outbound [SSP_COMPLETION Response](#) with STATUS set to IO_UNDERFLOW. See Section [8.3](#) for details.

A command underrun/overflow may or may not be considered an error depending on the CDB that is sent. For example, a command underrun/underflow is normal for INQUIRY CDB but is an error for a READ CDB.

11.6.4.9 RESPONSE Frame with Sense/Response Data Received

If the SPC 8x6G SSP initiator port receives a RESPONSE frame with sense or response data present, it notifies the host.

The controller notifies the host with an [SSP_COMPLETION Response](#) outbound IOMB with STATUS set to IO_COMPLETED (0x00000000) and PARAM set to non-zero. This indicates that an I/O request has completed with a SCSI status other than GOOD_STATUS, and may also indicate a task management completion. The SSP RESP IU field contains at least the first 24 bytes of an SSP response IU. The host should read SENSE DATA LENGTH (bytes [19:16] of SSP response IU) and RESPONSE DATA LENGTH (bytes [23:20] of SSP response IU) to know the actual total SSP response length.

See Section [8.3](#) for the full description of the [SSP_COMPLETION Response](#) outbound IOMB.

Depending on the information in the SSP RESP IU, the host may need to do a specific action.

11.6.4.10 Invalid RESPONSE Frame Received

If the SPC 8x6G SSP initiator port receives a RESPONSE frame with an incorrect length, it does not detect it. The controller does not validate all fields within an SSP Response IU (such as SENSE DATA LENGTH and RESPONSE DATA LENGTH).

The controller notifies the host with an [SSP_COMPLETION Response](#) outbound IOMB with STATUS set to IO_COMPLETED (0x00000000) and PARAM set to non-zero. This indicates that an I/O request has completed with a SCSI status other than GOOD_STATUS, and may also indicate a task management completion. The SSP RESP IU field contains at least the first 24 bytes of SSP response IU. The host should read SENSE DATA LENGTH (bytes [19:16] of the SSP response IU) and RESPONSE DATA LENGTH (bytes [23:20] of the SSP response IU) to know the actual total SSP response length.

See Section [8.3](#) for the full description of the [SSP_COMPLETION Response](#) outbound IOMB.

Depending on the information in the SSP RESP IU, the host may need to do a specific action.

11.6.5 SSP Transport Layer (Target Port)

11.6.5.1 XFER_RDY/RESPONSE Frame Received

If the SPC 8x6G SSP target port receives an XFER_RDY frame it sends an [SSP_COMPLETION Response](#) with STATUS set to IO_XFER_ERROR_UNEXPECTED_PHASE as described in Section [8.3](#).

If the SPC 8x6G SSP target port receives a RESPONSE frame, it discards the frame.

11.6.5.2 Vendor Specific Frame Received

If the SPC 8x6G SSP target port receives a vendor-specific SSP frame type, it notifies the host by a message indicating SSP frame received.

The controller notifies the host with an [SSP_REQUEST_RECEIVED Notification](#) outbound IOMB that contains the SSP vendor-specific IU. See Section 8.13 for the full description of the [SSP_REQUEST_RECEIVED Notification](#) outbound IOMB.

The host retrieves the payload of the frame and decodes the vendor-specific frame.

11.6.5.3 COMMAND Frame with Existing TAG Received

If the SPC 8x6G SSP target port receives COMMAND frame with the same tag as another COMMAND, it does not detect it. The controller forwards the frame to host.

The controller notifies the host with an [SSP_REQUEST_RECEIVED Notification](#) outbound IOMB that contains the SSP vendor-specific IU. See Section 8.13 for the full description of the [SSP_REQUEST_RECEIVED Notification](#) outbound IOMB.

After retrieving the payload of the frame, the host can detect duplicate tags with an existing request. As recommended by the SCSI standard, the host may send a RESPONSE with CHECK CONDITION with sense key set to ABORTED COMMAND and a sense code set to OVERLAPPED COMMAND ATTEMPTED.

11.6.5.4 Invalid COMMAND/TASK Frame Received

If the SSP target port receives a COMMAND/TASK frame with a TARGET PORT TRANSFER TAG other than 0xFFFF, the SPC 8x6G controller does not detect it. The controller forwards the frame to a host.

After retrieving the payload of the frame, the host can detect the invalid frame. As described by the SAS specification, the host may send a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME.

11.6.5.5 Unexpected DATA Frame Received

If the SPC 8x6G SSP target port receives a DATA frame with a command tag, when it is not expecting DATA frame, it discards the frame and notifies the host with an [SSP_COMPLETION Response](#) with the EVENT field set to IO_XFER_ERROR_UNEXPECTED_PHASE as described in Section 8.11.

11.6.5.6 DATA Frame with Unknown Tag

If the SPC 8x6G SSP target port receives DATA frame with unknown tag, it discards the frame.

11.6.5.7 Invalid DATA Frame Received

If the SPC 8x6G SSP target port receives an invalid DATA frame with one of the following errors, it discards the frame, and notifies the host with an error message as shown in [Table 374](#). The controller notifies the host with an [SSP_COMPLETION Response](#) outbound IOMB with EVENT set shown in [Table 374](#).

Table 374 SSP Target Invalid DATA Frame Received

Error Message	EVENT Setting
Too much read data (command overrun)	IO_OVERFLOW (0x00000002)
Information Unit too short	IO_UNDERFLOW if the DATA frame contains a complete frame header; otherwise, the packet will be silently dropped.
Offset error	IO_XFER_ERROR_OFFSET_MISMATCH (0x00000034)

See Section 8.11 for the full description of the [SSP_COMPLETION Response](#) outbound IOMB.

11.6.5.8 Initiator Response Timeout

The initiator response timer is the time between the time target transfers XFER_RDY frame and the time the target receives the entire valid DATA frame for the XFER_RDY frame.

The SPC 8x6G controller does not implement this timer.

The host may optionally setup a timer before sending an [SSP_TGT_IO_START Command](#) to send the XFER_RDY, and if the timer expires before the host receives an [SSP_COMPLETION Response](#) for the XFER_RDY frame, the host may generate a response error INITIATOR RESPONSE TIMEOUT to the initiator. See Sections 7.8 and 8.3 for details.

The host as a target may also rely on the initiator side to do the timeout recovery when the overall I/O operation does not complete successfully.

11.6.6 SMP Transport Layer

11.6.6.1 SMP Response Timeout

If the SPC 8x6G SMP initiator port detects a timeout for SMP response, it notifies the host with a message.

The controller notifies the host with an [SMP_COMPLETION Response](#) outbound IOMB with STATUS set to IO_ERROR_HW_TIMEOUT. See Section 8.4 for details.

The host can retry the SMP request later.

11.7 Detailed Description of SATA Error Conditions

This section describes SPC 8x6G SATA error conditions based on the layered model of SATA protocol:

- SATA physical layer
- SATA link layer
- SATA transport layer

11.7.1 SATA Physical Layer

11.7.1.1 SATA PHY Not Ready

The SPC 8x6G controller notifies the host with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_PHY_RESET_FAILED when the PHY fails to complete an OOB handshake or speed negotiation. See Section 8.2 for details. The host can directly read and reset the PHY reset failed counter using the [PHY Reset Failed Count Register](#) described in Section 10.6.10.

11.7.1.2 SATA Running Disparity Error

This error is handled in a similar manner to a SAS Running Disparity Error. See Section 11.6.1.3, “[SAS Running Disparity Error](#)” for details.

11.7.1.3 Code Violation/Invalid DWord

When the SPC 8x6G controller detects a code violation (which is the same as the invalid DWord for SAS error detection), either inside a FIS or outside a FIS, it increments the [Counter Configuration Register](#) described in Section 10.6.5 and notifies the host with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INVALID_DWORD. See Section 8.2.

The host can directly read and reset the code violation / invalid DWord counter using the [Code Violation Error Count Register](#) described in Section 10.6.8.

11.7.1.4 Loss of DWord Synchronization

This error is handled in a similar manner to the SAS Loss of DWord Synchronization error. See Section 11.6.1.4, “[Loss of DWord Synchronization](#)” for details.

11.7.2 SATA Link Layer

11.7.2.1 CRC Error

If a CRC error is detected after receiving the start of a FIS frame and before receiving an EOF, the SPC 8x6G controller detects the bad CRC, discards the frame, and sends an R_ERR to the remote SATA device. The SPC 8x6G notifies the host with a [SAS_HW_EVENT Notification](#) with the EVENT field set to IOP_EVENT_PHY_ERR_INBOUND_CRC. See Section 8.2, “[SAS_HW_EVENT Notification](#)” for details. The subsequent action differs for NCQ and non-NCQ modes:

- NCQ mode: If the SET_DEVICE_BITS FIS is received by the drive with the error bit set, the SPC 8x6G notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_ABORTED_NCQ_MODE. See Section 8.10 for details. The recovery action is to perform a READ_LOG_EXT followed by the [SATA_ABORT Command](#) described in Section 7.17.

- Non-NCQ mode: If the D2H FIS is received with the error bit set, the SPC 8x6G notifies the host with a [SATA_COMPLETION Response](#) with the STATUS set to IO_COMPLETED. See Section 8.9. Firmware returns the D2H FIS to the host.

If the SATA drive does not retry sending the frame, this error causes the FIS operation to fail/timeout. The host must do the normal recovery procedure for SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation).

11.7.2.2 WTRM Received Between SOF and EOF

If a Wait For Frame Termination (WTRM) primitive is received after an SOF is received and before the EOF is received (this may happen because of missing the detection of the EOF), the SPC 8x6G controller discards the frame and sends a R_ERR to the remote SATA device.

If the SATA drive does not resend the frame, this error causes the host-command timeout. The host must complete the normal recovery procedure for SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation).

11.7.2.3 SYNC Received Between SOF and EOF

If a SYNC is received after an SOF is received and before an EOF is received, the SPC 8x6G controller discards the frame.

If the SATA drive does not resend the frame, this error causes the FIS host command to time out. The host must complete the normal recovery procedure for SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation).

11.7.2.4 SYNC Received During FIS Transmission

If a SYNC is received during:

- The FIS transmission, the SPC 8x6G controller stops the transmission of the frame and notifies the host.
- H2D FIS transmission, the SPC 8x6G notifies the host with a [SATA_COMPLETION Response](#) with the STATUS field set to IO_XFER_ERROR_SATA_LINK_TIMEOUT.
- Data FIS transmission, the SPC 8x6G notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_PEER_ABORTED. The subsequent actions differ for NCQ and non-NCQ modes:
 - NCQ mode: If the SET_DEVICE_BITS FIS is received from the drive with the error bit set, the SPC 8x6G notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_ABORTED_NCQ_MODE. See Section 8.10 for details. The recovery action is to perform a READ_LOG_EXT followed by the [SATA_ABORT Command](#) described in Section 7.17.
 - Non-NCQ mode: If the D2H FIS is received with the error bit set, the SPC 8x6G notifies the host with a [SATA_COMPLETION Response](#) with the STATUS field set to IO_COMPLETED. See Section 8.9. Firmware returns the D2H FIS to the host.

11.7.2.5 R_ERR Received

The SPC 8x6G controller notifies the host upon receiving an R_ERR for any FIS that is transmitted to the drive. For an R_ERR that is received during:

- H2D FIS, the SPC 8x6G notifies the host with a [SATA_COMPLETION Response](#) with the STATUS field set to IO_XFER_ERROR_NAK RECEIVED. See Section [8.9](#) for details.
- Data FIS, the SPC 8x6G notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_NAK RECEIVED. See Section [8.10](#) for details.

11.7.2.6 Receiving R_OK/R_ERR Timeout

By default, the SPC 8x6G controller does not timeout when waiting to receive R_OK or R_ERR. It will wait indefinitely, resulting in the host detecting the timeout. In this case, the host must complete a normal recovery procedure for SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation). However, this behavior can be altered by enabling the STP_FRAME_TIMER and specifying a timeout value. See Section [10.7.11](#), “[Timer Enables Register](#)” for how to enable the timer and Section [10.5.3](#), “[Timer Control 2 Register](#)” for how to specify the STP_FRAME_TIMEOUT value.

If the timer is enabled, the SPC 8x6G will notify the host upon timer expiration. During H2D FIS, the SPC notifies the host with a [SATA_COMPLETION Response](#) with the STATUS field set to IO_XFER_ERROR_SATA_LINK_TIMEOUT. During data FIS, the SPC notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_PEER_ABORTED. See Sections [8.9](#) and [8.10](#) for details.

11.7.2.7 Receiving R_RDY Timeout

The SPC 8x6G handles R_RDY timeouts in a similar manner to R_OK or R_ERR timeouts. See Section [11.7.2.6](#), “[Receiving R_OK/R_ERR Timeout](#)” for details.

11.7.2.8 Unexpected SOF

When the SPC 8x6G controller receives an SOF before it transmits an R_RDY, it discards the frame and issues a SYNC to the remote device.

The host detects a timeout for the operation. The host must complete a normal recovery procedure for SATA I/O timeout (that is, abort the pending operation, issue a device reset and retry the operation).

11.7.2.9 PHY Not Ready During Transmission

The SPC 8x6G controller detects the link down condition during FIS transmission. It aborts the operation and the host is notified with an event or completion depending on the phase.

For a link down during:

- D2H FIS, the SPC 8x6G controller notifies the host with a [SATA_COMPLETION Response](#) with the STATUS set to IO_XFER_ERROR_PHY_NOT_READY. See Section [8.9](#) for details.
- Data FIS, the SPC 8x6G controller notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_XFER_ERROR_PHY_NOT_READY. See Section [8.10](#) for details.

11.7.3 SATA Transport Layer

11.7.3.1 Unrecognized FIS

If the SPC 8x6G controller receives an FIS with an unrecognized or invalid type, it discards the frame.

Note that the controller only supports SATA host mode, but does not support SATA device mode. So the received FIS here indicates the remote device response to the initial SPC 8x6G FIS sent to initiate an operation.

The host may detect a timeout for a SATA command. In this case, the host must perform the normal recovery procedure for a SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation).

11.7.3.2 Malformed FIS

If the SPC 8x6G controller detects a malformed FIS, for example an FIS with an incorrect length or incorrect FIS fields, it discards the frame. The host is not notified.

The host may detect a timeout for a SATA command. In this case, the host must perform the normal recovery procedure for a SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation).

11.7.3.3 PIO Data Overflows

If the SPC 8x6G controller detects that the received DATA for PIO read command is more than what is specified in PIO setup, it discards the last DATA FIS and sends an R_ERR to the remote device.

The controller notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_OVERFLOW followed by an outbound [SATA_COMPLETION Response](#) with the STATUS field set to IO_XFER_ERROR_DMA. See Sections [8.9](#) and [8.10](#) for details.

11.7.3.4 PIO Data Underruns

If a remote device does not send enough data for a PIO read command, the SPC 8x6G controller does not detect it.

The host detects a timeout for the operation. The host must do the normal recovery procedure for SATA I/O timeout (that is, abort the pending operation, issue a device reset, and retry the operation).

11.7.3.5 DMA Data Overflows

If the SPC 8x6G controller detects that received DATA for a DMA read command is more than specified in DMA setup, it detects it and notifies the host.

The controller notifies the host with a [SATA_EVENT Notification](#) with the EVENT field set to IO_OVERFLOW. Subsequently, if the driver sends D2H FIS, the SPC 8x6G notifies the host with an outbound [SATA_COMPLETION Response](#) with the STATUS field set to IO_XFER_ERROR_DMA. See Sections [8.9](#) and [8.10](#) for details.

11.7.3.6 DMA Data Underruns

If the remote device does not send enough data for a DMA read command, the SPC 8x6G controller does not detect it. This results in a host command timeout. However, if D2H FIS is received before all the data is received for the DMA read command, the SPC 8x6G notifies the host with an outbound [SATA_COMPLETION Response](#) with the STATUS field set to IO_UNDERFLOW as described in Section [8.9](#).

11.7.3.7 Receive FIFO Overflows

The SPC 8x6G controller sends a SATA HOLD to prevent a FIFO overflow. If the remote device does not honor the SPC 8x6G HOLD by replying with HOLDA, and instead sends more frames, a SPC 8x6G FIFO overflow occurs.

11.7.3.8 Transmit FIFO Underruns

When the SPC 8x6G controller detects a transmit FIFO underrun condition, it continues sending a HOLD primitive to the remote device until a data frame is available to send.

This is not a real error condition as this is done by the SPC 8x6G hardware automatically. The host does not detect it.

11.7.3.9 Device to Host FIS with Error Information

The SPC 8x6G controller forwards the Device-to-Host FIS to the host as part of the [SATA_COMPLETION Response](#) outbound IOMB. See Section [8.9](#) for the full description of the [SATA_COMPLETION Response](#) outbound IOMB.

Glossary

Acronym	Definition
AAP1	Application Acceleration Processor (MIPS 34K CPU)
ACA	Auto Contingent Allegiance
ACK/NAK	Acknowledged/Not Acknowledged
AXI	Advanced eXtensible Interface
BAR	Base Address Register
BDMA	Block DMA
CDB	Command Descriptor Block
CI	Consumer Index
CMOS	Complementary Metal Oxide Semiconductor
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
EBC	Error Block Count
ECC	Error Correcting Code
EEPROM	Electrically Erasable Programmable Read-Only Memory
EOAF	End of Address Frame
EOB	End of Buffer
EOF	End of Frame
ERA	Exclusive Read Access
ESGL	Extended Scatter/Gather List
ESPRAM	External Scratchpad RAM
FCBGA	Flip-Chip Ball Grid Array
FIFO	First In, First Out
FIS	Frame Information Structure
FP	Frame Processor
FPDMA	First Party DMA
GSM	Global Shared Memory
GST	General Status Table
HBA	Host Bus Adapter
HDD	Hard Disk Drive
HSST	Hardened SAS/SATA Transport
ILA	Image Loader Agent
ISR	Interrupt Service Routine
ISTR	Initialization String
IOMB	I/O Message Buffer
IOP	Input/Output Processor (MIPS34K CPU)
IQ	Inbound Queue
IQCT	Inbound Queue Configuration Table

Acronym	Definition
iSCSI	Internet SCSI
IU	Information Units
L/A	Length/Address
LBA	Logical Block Address
LSB	Least Significant Bit
LUN	Logical Unit Number
MBIC	MIPS34K Bridge and Interrupt Controller
MF	Message Frame
MFA	Message Frame Address
MPI	Message Passing Interface
MPS	Maximum Payload Size
MSB	Most Significant Bit
MSGU	Message Unit (Firmware)
MXCBI	MBIC XCBI
NAK	Negative Acknowledgment
NCQ	Native Command Queuing
NMI	Non-maskable Interrupt
NVMD	NVM Device
OCP	Open Core Protocol
OOB	Out of Band
OQ	Outbound Queue
OQCT	Outbound Queue Configuration Table
OSSP	Octal SAS/SATA Port
PBA	Physical Block Address
PBC	Peripheral Bus Controller
PCIe	PCI Express
PCS	Processor Complex System
PI	Producer Index
PIO	Programmed Input/Output
POST	Power-on Self Test
PMIC	PCI/Memory Interface Controller
RISC	Reduced Instruction Set Computer
SAM	SCSI Architecture Model
SASIDAF	SAS Identify Address Frame
SATA	Serial ATA
SEEPROM	Serial Electrically Erasable Programmable Read-Only Memory
SERDES	Serializer/Deserializer
SES	SCSI Enclosure Services
SGL	Scatter/Gather List

Acronym	Definition
SGPIO	Serial General Purpose Input/Output
SMP	Serial Management Protocol
SOAF	Start of Address Frame
SOF	Start of Frame
SPBC	SPC Peripheral Bus Controller
SPI	Serial Peripheral Interface
SRAM	Serial RAM
SSP	Serial SCSI Protocol
SSPA	SAS/SATA Port Adapter
SSPIU	SSP Information Unit
SSPL	SAS/SATA Physical Layer
STP	Serial ATA Tunneled Protocol
TLR	Transport Layer Retry
TRGT	Target
TWI	Two-wire Interface
UART	Universal Asynchronous Receiver / Transmitter
UDT	User Defined Tag
UUM	UDT Update Mask
UVM	UDT Verification Mask
VPD	Vital Product Data
WSTRB	Write Strobes
XCBI	Extended Common Bus Interface

End of Document

Downloaded [controlled] by Mark Salyzyn of Xyratex on Monday, 28 March, 2011 05:50:14 AM